Explainable Matrix – Visualization for Global and Local Interpretability of Random Forest Classification Ensembles

Mário Popolin Neto and Fernando V. Paulovich, *Member, IEEE*

Abstract—Over the past decades, classification models have proven to be essential machine learning tools given their potential and applicability in various domains. In these years, the north of the majority of the researchers had been to improve quantitative metrics, notwithstanding the lack of information about models' decisions such metrics convey. This paradigm has recently shifted, and strategies beyond tables and numbers to assist in interpreting models' decisions are increasing in importance. Part of this trend, visualization techniques have been extensively used to support classification models' interpretability, with a significant focus on rule-based models. Despite the advances, the existing approaches present limitations in terms of visual scalability, and the visualization of large and complex models, such as the ones produced by the Random Forest (RF) technique, remains a challenge. In this paper, we propose *Explainable Matrix (ExMatrix)*, a novel visualization method for RF interpretability that can handle models with massive quantities of rules. It employs a simple yet powerful matrix-like visual metaphor, where rows are rules, columns are features, and cells are rules predicates, enabling the analysis of entire models and auditing classification results. ExMatrix applicability is confirmed via different examples, showing how it can be used in practice to promote RF models interpretability.

Index Terms—Random forest visualization, logic rules visualization, classification model interpretability, explainable artificial intelligence

1 Introduction

Imagine a machine learning classification model for cancer prediction with 99% accuracy, prognosticating positive breast cancer for a specific patient. Even though we are far from reaching such level of precision, we (researchers, companies, among others) have been trying to convince the general public to trust classification models, using the premise that machines are more precise than humans [15]. However, in most cases, yes or no are not satisfactory answers. A doctor or patient inevitably may want to know why positive? What are the determinants of the outcome? What are the changes in patient records that may lead to a different prediction? Although standard instruments for building classification models, quantitative metrics such as accuracy and error cannot tell much about the model prediction, failing to provide detailed information to support understanding [38].

We are not advocating against machine learning classification models, since there is no questioning about their potential and applicability in various domains [10, 20]. The point is the acute need to go beyond tables and numbers to understand models' decisions, increasing trust in the produced results. Typically, this is called model interpretability and has become the concern of many researchers in recent years [11, 60]. Model interpretability is an open challenge and opportunity for researchers [20] and also a government concern, as the *European General Data Protection Regulation* requires explanations about automated decisions regarding individuals [11, 27, 39].

Model interpretability strategies are typically classified as global or local approaches. Global techniques aim at explaining entire models, while the local ones give support for understanding the reasons for the classification of a single instance [11, 19]. In both cases, interpretability can be attained using inherent interpretable models such as Decision Trees, Rules Sets, and Decision Tables [31], or through surrogates, where black-box models, like Artificial Neural Networks or Support Vector Machines, are approximated by rule-based interpretable models [11, 27]. The key idea is to transform models into logic rules, using them as a mechanism to enable the interpretation of a model and its decisions [17, 26, 35, 42, 46].

Recently, visualization techniques have been used to empower the process of interpreting rule-based classification models, particularly Decision Tree models [17, 47, 57, 61]. In this case, given the inherent nature of these models, the usual adopted visual metaphors focus on revealing tree structures, such as the node-link diagrams [25, 42, 61]. However, node-link structures are limited when representing logic rules [22, 29, 37], and present scalability issues, supporting only small models with few rules [25, 48, 61]. Matrix-like visual metaphors have been used [17,42] as an alternative, but visual scalability limitations still exist, and large and complex models cannot be adequately visualized, such as the Random Forests [6, 7]. Among rule-based models, Random Forests is one of the most popular techniques given their simplicity of use and competitive results [6]. However, they are very complex entities for visualization since multiple Decision Trees compose a model, and, although attempts have been made to overcome such a hurdle [61], the visualization of entire models is still an open challenge.

In this paper, we propose *Explainable Matrix (ExMatrix)*, a novel method for Random Forest (RF) interpretability based on the visual representations of logic rules. ExMatrix supports global and local explanations of RF models enabling tasks that involve the overview of models and the auditing of classification processes. The key idea is to explore logic rules by demand using matrix visualizations, where rows are rules, columns are features, and cells are rules predicates. ExMatrix allows reasoning on a considerable number of rules at once, helping users to build insights by employing different orderings of rules/rows and features/columns, not only supporting the analysis of subsets of rules used on a particular prediction but also the minimum changes at instance level that may change a prediction. Visual scalability is addressed in our solution using a simple yet powerful compact representation that allows for overviewing entire RF models while also enables focusing on specific parts for details on-demand. In summary, the main contributions of this paper are:

- A new matrix-like visual metaphor that supports the visualization of RF models;
- A strategy for Global interpretation of large and complex RF models supporting model overview and details on-demand; and
- A strategy to promote Local interpretation of RF models, supporting auditing models' decisions.

- *M. Popolin Neto is with Federal Institute of São Paulo (IFSP) and University of São Paulo (USP), Brazil. E-mail: mariopopolin@ifsp.edu.br*
- *F. V. Paulovich is with Dalhousie University, Canada, and University of São Paulo (USP), Brazil. E-mail: paulovich@dal.ca*

Manuscript received 30 Apr. 2020; revised 31 July 2020; accepted 14 Aug. 2020. Date of publication 13 Oct. 2020; date of current version 15 Jan. 2021. Digital Object Identifier no. 10.1109/TVCG.2020.3030354

2 Related Work

Typically, visualization techniques aid in classification tasks in two different ways. One is on supporting parametrization and labeling processes aiming to improve model performance [3, 18, 30, 34, 38, 53, 55, 57]. The other is on understanding the model as a whole or the reasons for a particular prediction. In this paper, our focus is on the latter group, usually named model interpretability.

Interpretability techniques can be divided into pre-model, in-model, or post-model strategies, regarding support to understand classification results before, during, or after the model construction [11]. Pre-model strategies usually give support to data exploration and understanding before model creation [11, 14, 41, 43]. In-model strategies involve the interpretation of intrinsically interpretable models, such as Decision Trees, and post-model strategies concerns interpretability of complete built models, and they can be model-specific [44, 59] or model-agnostic [17, 26, 42, 46]. Both in-model and post-model approaches aim to provide interpretability by producing global and/or local explanations [19].

2.1 Global Explanation

Global explanation techniques produce overviews of classification models aiming at improving users' trust in the model [45]. For inherently interpretable models, the global explanation is attained through visual representations of the entire model. For more complex non-interpretable black-box models, such as Artificial Neural Networks or Support Vector Machines, interpretability can be attained through a surrogate process where such models are approximated by interpretable ones [17, 28, 42]. Decision Trees [9, 40, 54] are commonly used as surrogate models [17, 28], and whether a surrogate or a classification model per se, the most common visual metaphor for global explanation is the node-link [42, 61], such as the BaobaView technique [57]. The node-link metaphor's problem is scalability [25, 48, 61], mainly when it is used to create visual representations for Random Forests, limiting the model to be small in number of trees [51]. Creating a scalable visual representation for an entire Random Forest model, presenting all decision paths (root node to leaf node paths), remains a challenge even with a considerably small number of trees [38].

Although the node-link metaphor is the straightforward representation for Decision Trees, logic rules extracted from decision paths have also been used to help on interpretation [37]. Indeed, disjoint rules have shown to be more suitable for user interpretation than hierarchical representations [33], and a user test comparing the node-link metaphor with different logic rule representations, showed that Decision Tables [31] (rules organized into tables) offers better comprehensibility properties [22, 29]. Nonetheless, this strategy uses text for representing rules having as drawback model size [22]. Similarly to Decision Tables, our method does not lean on the hierarchical property of Decision Trees. However, instead of using text to represent logic rules, we used a matrix-like visual metaphor, where rows are rules, columns are features, and cells are rules predicates, capable of displaying a much larger number of rules than the textual representations.

The idea of using a matrix metaphor to present rules is not new [17, 42], and it has been used before by the RuleMatrix technique [42]. RuleMatrix is a model-agnostic approach to induce logic rules from black-box models, presenting rules in rows, features in columns, and predicates in cells using histograms. As data histograms require a certain display space to support human cognition, the number of rules displayed at once is reduced. Therefore, not being able to present entire or even parts of Random Forest models (notice that their focus is the visualization of surrogate rules, not models). Our approach also uses a matrix metaphor; however, we employ a simpler icon (colored rectangular shape) for the matrix cells, mapping different rule properties (e.g., predicates, class, and others), considerably improving the scalability of the visual representation. Besides the recognized scalability of matrix visualization and custom cells [1, 2, 4], rows and columns order is an important principle [4, 12, 13, 58], and in our approach rules and features can be organized using different criteria, promoting analytical tasks not supported by the RuleMatrix, such as the holistic analysis of Random Forest models through complete overviews. Worthy mentioning that different from usual matrix visual metaphors for trees and graphs that focus on nodes [4, 25], our approach focus on decision paths, which is the object of analysis on Decision Trees [22, 29, 37], so representing a different concept.

2.2 Local Explanation

Unlike the model overview of global explanations, local explanation techniques focus on a particular instance classification result [46, 61], aiming to improve users' trust in the prediction [45]. As in global strategies, local explanations can be provided using inherently interpretable models or using surrogates of black-boxes [26, 46, 52]. In general, local explanations are constructed using the logic rule applied to classify the instance along with its properties (e.g., coverage, certainty, and fidelity), providing additional information for prediction reasoning [33, 42].

One example of a visualization technique that supports local explanation is the RuleMatrix [42]. RuleMatrix was applied to support the analysis of surrogate logic rules of Artificial Neural Networks and Support Vector Machine models. Local explanations are taken by analyzing the employed rules, observing the instance features values coupled with rules predicates and properties. Another interactive system closely related to our method is the iForest [61], combining techniques for Random Forest models local explanations. The iForest system focuses on binary classification problems, and for each instance, it allows the exploration of decision paths from Decision Trees using multidimensional projection techniques. A summarized decision path is built and displayed as a node-link diagram by selecting decision paths of interest (circles in the projection).

As discussed before, node-link diagrams are prone to present scalability issues. Although iForest reduces the associate issues by summarizing similar decision paths, it fails to present the overall picture of Random Forest classification models' voting committees. Our approach shows the voting committee by displaying all rules (decision paths) used by a model when classifying a particular instance, allowing insights into the feature space and class association by ordering rules and features in different ways. Also, our approach can be applied to multi-class problems, not only binary classifications, and, as iForest, it supports counterfactual analysis [24, 36] by displaying the rules that, with the smallest changes, may cause the instance under analysis to switch its final classification.

3 ExMatrix

In this section, we present *Explainable Matrix (ExMatrix)*, a visualization method to support Random Forest global and local interpretability.

3.1 Overview

To create a classifier, classification techniques take a labelled dataset $X = \{x_1, ..., x_N\}$ with N instances and their classes $Y = \{y_1, ..., y_N\}$, where $y_n \in C = \{c_1, ..., c_J \geq 2\}$ and x_n consists of a vector $x_n = [x_n^1, ..., x_n^M]$ with M features $F = \{f_1, ..., f_M\}$ values, and build a mathematical model to compute a class y_n when new instances $x_n \notin X$ are given as input. In this process, X is usually split into two different sets, one X_{train} to build the model and one X_{test} to test it. The existing techniques have adopted many different strategies to build a classifier. The Random Forest (RF) is an ensemble approach that creates multiple Decision Tree (DT) models $DT_1, ..., DT_K$ of randomly selected subsets of features and/or training instances, and combines them to classify an instance using a voting strategy [6, 7, 9, 54]. Therefore, a RF model is a collection of decision paths, belonging to different DTs, combined to classify an instance.

Aiming at supporting users to examine RF models and enable results audit, ExMatrix presents the decision paths extracted from DTs as logic rules using a matrix visual metaphor, supporting global and local explanations. ExMatrix arranges logic rules $R = \{r_1, ..., r_Z\}$ as rows, features $F = \{f_1, ..., f_M\}$ as columns, and rule predicates $r_z = [r_z^1, ..., r_z^M]$ as cells, inspired by similar user-friendly and powerful matrix-like solutions [12, 13, 58]. Fig. 1 depicts our method overview, composed mainly of two steps. One involving the vector rules extraction, where all decision paths of each DT_k in the RF model are converted into vectors, and a second one where these vectors are displayed using a matrix

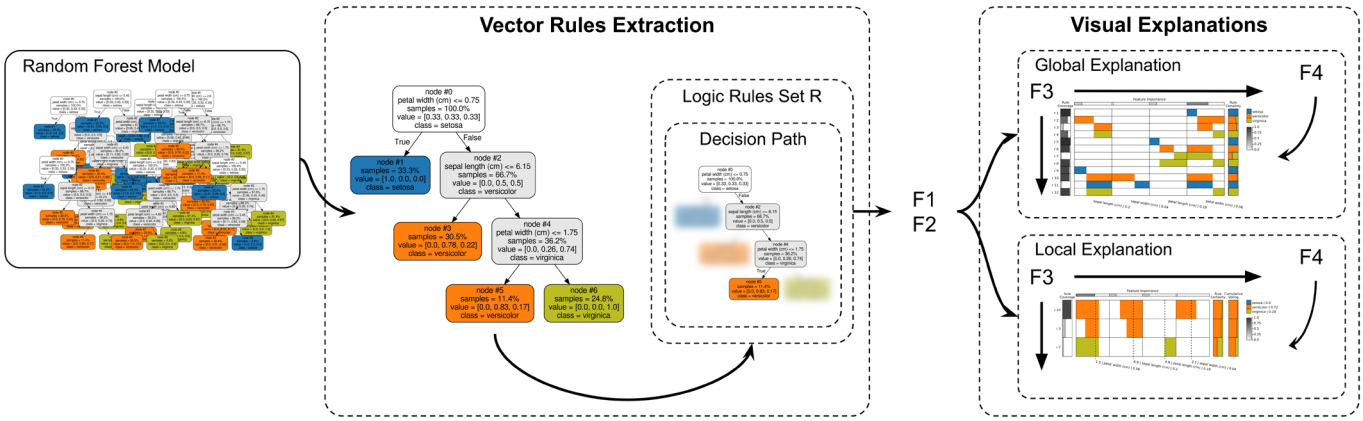


Fig. 1. *Explainable Matrix (ExMatrix)* overview. ExMatrix is composed of two main steps. In the first, decision paths of the RF model under analysis are converted into logic rules. Then, in the second, these rules are displayed using a matrix metaphor to support global and local explanations.

metaphor to support explanations. The next sections detail these steps, starting with the vector rule extraction process.

3.2 Vector Rules Extraction

As mentioned, ExMatrix first step involves the transformation of each decision path, the path from a DT root node to a leaf node, into a vector rule representing the features' intervals for which the decision path is true. The resulting vectors present dimensionality equal to the number of features M, with coordinates composed of pairs representing the features' minimum and maximum interval values. In more mathematical terms, this process transforms, for every tree DT_k, each decision path $p_{(o,d)}$ (from the root node o to the leaf node d) into a disjoint logic rule (vector) r_z. Let $p_{(o,d)} = \{(f_o \bigotimes \theta_o), ..., (f_v \bigotimes \theta_v)\}$ denotes a decision path, where each node i contains a logic test $\bigotimes \in \{\text{“}\leq\text{”}, \text{“}>\text{”}\}$ bisecting the feature f_i using a threshold $\theta_i \in \mathbb{R}$, and that the node v is the parent of the leaf node d [61]. To convert $p_{(o,d)}$ into a vector rule $r_z = [r_z^1, ..., r_z^M]$, each element $r_z^m = \{\alpha_z^m, \beta_z^m\}$ is computed representing the intervals covered by $p_{(o,d)}$ if and only if $f^m \in p_{(o,d)}$. Otherwise, $r_z^m = \varnothing$. Considering $f^m \in p_{(o,d)}$, the lower limit α_z^m is the maximum $\theta_i \in p_{(o,d)}$ for the feature f^m and logic test $\bigotimes = \text{“}>\text{”}$. If such combination does not exist in $p_{(o,d)}$, α_z^m is set to the minimum value of feature f^m in X, that is

$$\alpha_z^m = \begin{cases} \max(\theta_i | f_i = f^m, \bigotimes = \text{“}>\text{”}) & \text{if } (f_i = f^m > \theta_i) \in p_{(o,d)} \\ \min(x^m | x^m \in X) & \text{Otherwise.} \end{cases}$$

Similarly, the upper limit β_z^m is the minimum $\theta_i \in p_{(o,d)}$ for the feature f^m and logic test $\bigotimes = \text{“}\leq\text{”}$. If such combination does not exist in $p_{(o,d)}$, β_z^m is set to the maximum value of feature f^m in X, that is

$$\beta_z^m = \begin{cases} \min(\theta_i | f_i = f^m, \bigotimes = \text{“}\leq\text{”}) & \text{if } (f_i = f^m \leq \theta_i) \in p_{(o,d)} \\ \max(x^m | x^m \in X) & \text{Otherwise.} \end{cases}$$

Beyond predicates, three other properties are extracted for each logic rule r_z, being certainty, class, and coverage. The rule certainty r_z^{cert} is a vector of probabilities for each class $c_j \in C$, obtained from the decision path (leaf node value). The rule class r_z^{class} is the $c_j \in C$ with the highest probability on the rule certainty r_z^{cert}. The rule coverage r_z^{cov} is the number of instances in X_{train} of class r_z^{class} for which r_z is valid divided by the total number of instance of r_z^{class} in X_{train}. The vector rules extraction process results in a set of disjoint logic rules $R = \{r_1, ..., r_Z\}$, where each rule r_z classifies an instance x_n belonging to class r_z^{class} if its predicates $r_z = [r_z^1, ..., r_z^M]$ are all true for the feature values in x_n.

As an example of vector rule extraction, consider the zoomed DT in Fig. 1 from a RF for the Iris dataset [21], with 150 instances in three classes $C = \{setosa, versicolor, virginica\}$ and 4 features $F = \{sepal\ length, sepal\ width, petal\ length, petal\ width\}$. From this tree, the decision path $p_{(\#0,\#5)}$ is transformed into the vector rule $r_3 = [\{6.15, 7.9\}, \varnothing, \varnothing, \{0.75, 1.75\}]$ with $r_3^{class} = versicolor$, since rule certainty equals to $r_3^{cert} = [0.0, 0.83, 0.17]$ (leaf node #5 value), indicating that r_3 is valid for 83% of the *versicolor* instances and 17% of *virginica* instances in X_{train}. The rule coverage $r_3^{cov} = 0.28$ as r_3 is valid for 10 out of 35 *versicolor* instances in X_{train}.

3.3 Visual Explanations

Once the vector rules are extracted, they are used to create the matrix visual representations for global and local interpretation. To guide our design process we adopted the iForest design goals (G1 - G3) [61] and the RuleMatrix target questions (Q1 - Q4) [42] summarized on Table 1. These goals and questions consider classification model reasoning beyond performance measures (e.g., accuracy and error), focusing on the model internals. For global explanations, where the focus is an overview of a model, ExMatrix displays feature space ranges and class associations (**G1** and **Q1**), and how reliable these associations are (**Q2**). For local explanations, where the focus is the classification of a particular instance x_n, ExMatrix allows the analysis of x_n values and features space ranges that resulted into the assigned class y_n (**G2** and **Q3**), and the inspection of the changes in x_n that may lead to a different classification (**G3** and **Q4**).

Table 1. ExMatrix design goals.

Global	Local
G1 Reveal the relationships between features and predictions [61].	**G2** Uncover the underlying working mechanisms [61].
Q1 What knowledge has the model learned? [42]	**G3** Provide case-based reasoning [61].
Q2 How certain is the model for each piece of knowledge? [42]	**Q3** What knowledge does the model utilize to make a prediction? [42]
	Q4 When and where is the model likely to fail? [42]

ExMatrix implements these goals using a set of four functions:

F1 – Rules of Interest. Function $R' = f_{rules}(R, ...)$ returns a subset of rules of interest $R' \subseteq R$. For global explanations $f_{rules}(R, ...)$ returns the entire vector rules set $R' = R$ or a subset $R' \subset R$ defined by the user, while for local explanations $f_{rules}(R, x_n, ...)$ returns a subset $R' \subset R$ related to a given instance x_n.

F2 – Features of Interest. Function $F' = f_{features}(R', ...)$ returns features of interest $F' \subseteq F$ considering a set of rules of interest R'. For global explanations $f_{features}(R', ...)$ returns all features used by the RF model, whereas for local explanations

$f_{features}(R', x_n, ...)$ returns the features used to classify a given instance x_n.

F3 – Ordering. Function $L' = f_{ordering}(L, criteria, \ldots)$ returns an ordered version L' of a input set L following a given criterion, where L can be rules R' or features F'. This is used for both global and local explanations aiming at revealing patterns, a key property in matrix-like visualizations [12, 13, 58], where rows and columns can be sorted in different ways, following, for instance, elements properties [32] or similarity measures [5, 23, 49, 56].

F4 – Predicate Icon. Function $f_{icon}(r_z^m, \ldots)$ returns a cell icon (visual element) for a predicate r_z^m of the rule r_z and feature f_m. For global and local explanations, a cell icon is a color-filled rectangular element, allowing our visual metaphor to display a substantial number of logic rules at once. This is an important aspect since matrix-like visualizations can display a massive number of rows and columns relying on such icons not requiring many pixels [12].

Fig. 1 shows how these four functions are used in conjunction to build the visual representations for global and local interpretation. Functions **F1** and **F2** are used to select and map rules and features of interest. Function **F3** is used to change the rows and columns order to help in finding interesting patterns, and function **F4** is used to derive the predicate icon that can vary depending on the type of interpretation task (global or local). In the next section, we detail how these functions are used to build ExMatrix visual representations.

3.3.1 Global Explanation (GE)

Our first visual representation is an overview of RF models called *Global Explanation (GE)*. To build this matrix, $R' = f_{rules}(R, \ldots)$ returns all logic rules R or a subset $R' \subset R$ defined by the user, and $F' = f_{features}(R', \ldots)$ returns all features used by at least one rule $r_z \in R'$. As previously explained, matrix rows represent logic rules, columns features, and cells rules predicates (icons). Rows and columns can be ordered using different criteria ($L' = f_{ordering}(L, criteria, \ldots)$). The rows can be ordered by rules' coverage, certainty, class & coverage, and class & certainty, while columns can be ordered by feature importance, calculated using the Mean Decrease Impurity (MDI) [8].

For the ExMatrix GE visualization, the matrix cell icon representing the rule predicate r_z^m consists of a rectangle ($f_{icon}(r_z^m, \ldots)$) colored according to the rule class r_z^{class}, positioned and sized inside the matrix cell proportional to the predicate limits $\{\alpha_z^m, \beta_z^m\}$, where the left side of the matrix cell represents the value $\min(x^m | x^m \in X)$ and the right side $\max(x^m | x^m \in X)$ (goals **G1** and **Q1**). The cell background not covered by the predicate limits can be either white or be filled using a less saturated color. If no predicate is present, the matrix cell is left blank.

Rules and features properties are also exhibited using additional rows and columns (goal **Q2**). The rule coverage r_z^{cov} is shown using an extra column on the left side of the table with cells' color (grayscale) and fill proportional to the coverage. The rules certainty r_z^{cert} is shown in an extra column in the right side of the table with cells split into colored rectangles with sizes proportional to the probability of the different classes. The feature importance is shown in an extra row on the top of the table with cells' color (grayscale) and fill proportional to the importance. Also, labels are added below the matrix, combining feature name and importance value.

Fig. 2 presents a ExMatrix GE visualization of a RF model for the Iris dataset with 3 trees with maximum depth equals to 3. In this example, the rows (rules) are ordered by extraction order, and the columns (features) follows the dataset order. The logic rule $r_3 = [\{6.15, 7.9\}, \varnothing, \varnothing, \{0.75, 1.75\}]$ extracted from the decision path $p_{(\#0,\#5)}$ (see Fig. 1) is zoomed in. It is colored in orange since this is the color we assign to the *versicolor* class and it classifies 83% of the training instances as belonging to this class (17% belonging to *virginica*). Also, its coverage is $r_3^{cov} = 0.28$.

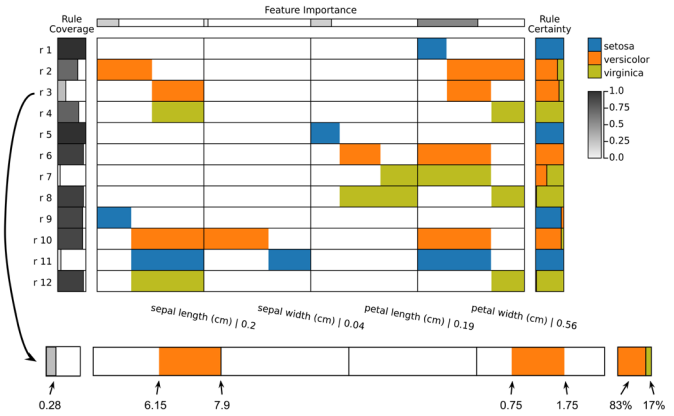


Fig. 2. ExMatrix Global Explanation (GE) of a RF model for the Iris dataset containing 3 trees with maximum depth equal to 3. Rows represent logic rules, columns features, and matrix cells the predicates. Additional rows and columns are also used to represent rule coverage and certainty. One matrix row is highlighted to exemplify how the rules' information is transformed into icons.

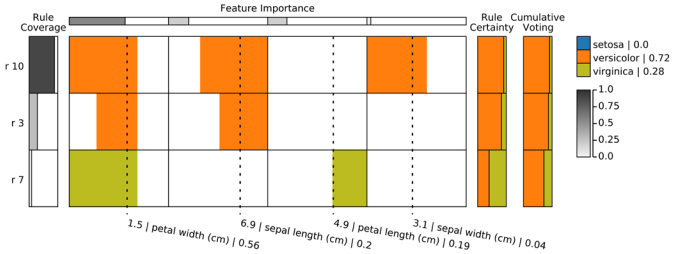


Fig. 3. ExMatrix Local Explanation showing the Used Rules (LE/UR) visualization. Three rules are used by the RF committee to classify a given instance as belonging to the *versicolor* class with 72% of probability. The dashed line in each column indicates the features' values of the instance.

3.3.2 Local Explanation Showing the Used Rules (LE/UR)

The second visual representation, called *Local Explanation Showing the Used Rules (LE/UR)*, is a matrix to help in auditing the results of a RF model providing explanations for the classification of a given instance x_n. In this process, $R' = f_{rules}(R, x_n)$ returns all logic rules used by the model to classify x_n (goals **G2** and **Q3**). As in the ExMatrix GE visualization, $F' = f_{features}(R')$ returns all features used by logic rules R', $f_{icon}(r_z^m, X)$ returns a cell icon representing predicates limits, and $f_{ordering}(L, criteria)$ can order rules R' by coverage, certainty, class & coverage, and class & certainty, and features F' by importance.

In addition to the coverage and certainty columns, in the ExMatrix LE/UR visualization, an extra column is added to represent the committee's cumulative voting. In this column, the cell at the i^{th} row is split into colored rectangles with sizes proportional to the different classes' probability considering only the first i rules. In this way, given a matrix order (e.g., based on the rule coverage), it is possible to see from what rule the committee reaches a decision that is not changed even if the remaining rules are used to classify x_n (indicated by a black line). Notice that this column's last cell always represents the committee's final decision regardless of rule ordering.

Fig. 3 presents the ExMatrix LE/UR representation for instance $x_{13} = [6.9, 3.1, 4.9, 1.5]$. We use the same RF model of Fig. 2 with 3 trees, so the RF committee uses 3 rules in the classification. The resulting matrix rows are ordered by rule coverage and columns by feature importance. The (optional) dashed line in each column indicates the values of the features of instance x_{13}. According to the committee, the probability of x_{13} to be *versicolor* is 72% and 28% to be *virginica*. Most of the *virginica* probability comes from the rule r_7, which holds the lowest coverage.

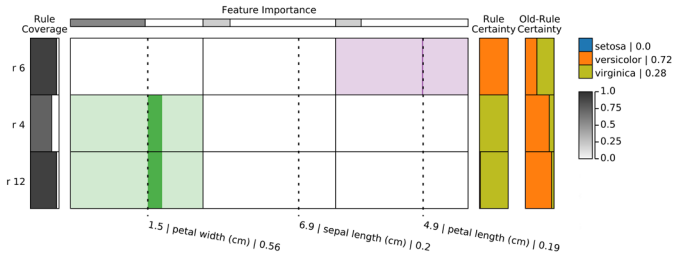


Fig. 4. ExMatrix Local Explanation Showing Smallest Changes (LE/SC) visualization. Three rules with the smallest change to make the DTs to change class decisions are displayed. The rule in the first row presents the smallest change. Small perturbations may change the RF classification decision.

3.3.3 Local Explanation Showing Smallest Changes (LE/SC)

Our final matrix representation, called *Local Explanation Showing Smallest Changes (LE/SC)*, is also designed to support results audit when classifying a given instance x_n. In this visualization, for each DT_k in the RF model, we display the rule requiring the smallest change to make DT_k to change the classification of x_n. Let r_z be the rule extracted from DT_k that is true when classifying x_n, in this process we seek for the rule r_e from DT_k with $r_e^{class} \neq r_z^{class}$ that presents the minimum summation of changes to the values of x_n that makes r_e true and r_z false, that is, $\Delta_{(r_e,x_n)} = \sum_{m=1}^{M}(\Delta^m_{(r_e,x_n)})$, where

$$\Delta^m_{(r_e,x_n)} = \begin{cases} \frac{\min(|\alpha_e^m - x_n^m|, |\beta_e^m - x_n^m|)}{|\max(x^m | x^m \in X_{train}) - \min(x^m | x^m \in X_{train})|} & \text{if } x_n^m \notin \{\alpha_e^m, \beta_e^m\} \\ 0 & \text{Otherwise.} \end{cases}$$

Using this formulation, function $R' = f_{rules}(R, x_n)$ returns the list of logic rules that can potentially change the classification process outcome requiring the lowest changes (goals **G3** and **Q4**), and function $F' = f_{features}(R', x_n)$ returns the features used by the rules in R'. Beyond the ordering criteria for rules and features previously discussed, function $f_{ordering}(L, criteria)$ also allows ordering using the change summation $\sum_{m=1}^{M}(\Delta^m_{(r_e,x_n)})$. Finally, function $f_{icon}(r_e^m, x_n)$ returns a rectangle positioned and sized proportional to the change $\Delta^m_{(r_e,x_n)}$, with positive changes colored in green and negative in purple, with the cell matrix background filled using a less saturated color. If $\Delta^m_{(r_e,x_n)} = 0$, the cell matrix is left blank. To help understand the class swapping, we add another column to the right of the table indicating the classification returned by the original rule r_z, showing the difference to the similar rule r_e that cause the DT_k to change prediction.

Fig. 4 shows the ExMatrix LE/SC visualization for instance $x_{13} = [6.9, 3.1, 4.9, 1.5]$ from the same RF model of Fig. 2. Features F' are ordered by importance and rules by change sum. The dashed lines represent the instance x_{13} values. As an illustration, rule r_6 presents the smallest change in the feature "petal length" to replace a rule of majority class *virginica* for a rule of class *versicolor*, potentially increasing the RF original outcome of 72% for class *versicolor* on instance x_{13}.

4 Results and Evaluation

In this section, we present and evaluate our method through a use-case [1] discussing the proposed features, two usage-scenarios [2][3] showing ExMatrix being used to explore RF models, finishing with a formal user test. All datasets employed in this section were downloaded from the *UCI Machine Learning Repository* [16], and the ExMatrix implementation is publicly available as a Python package at `https://pypi.org/project/exmatrix/` to be used in association with the most popular machine learning packages.

4.1 Use Case: Breast Cancer Diagnostic

In this use case, we utilize the *Wisconsin Breast Cancer Diagnostic (WBCD)* dataset to discuss how to use ExMatrix global and local explanations to analyze RF models. The WDBC dataset contains samples of breast mass cells of $N = 569$ patients, 357 classified as benign (B) and 212 as malignant (M), with $M = 30$ features (cells properties). The RF model used as example was created randomly selecting 70% of the instances for training and 30% for testing and setting the number of DTs to $K = 128$, not limiting their depths. The result is a model with 3,278 logic rules, 25.6 rules per DT, and an accuracy of 99%.

An overview of this model is presented in Fig. 5(a) using the ExMatrix GE representation (see Sect. 3.3.1). In this visualization, rules are ordered by coverage and features by importance. Using this ordering scheme, it is possible to see that "concave mean", "area worst", and "radius worst" are the three most important features, whereas "smoothness std", "texture std", and "fractal dimension mean" are less important, and that the RF model used all 30 features. Also, taking only the high coverage rules and features with more importance ("concave mean' to "radius mean"), some patterns in terms of predicate ranges emerge. To help verify these patterns, low-coverage rules can be filtered out, resulting in a new visualization containing only high-coverage rules. Fig. 5(b) presents the resulting filtered visualization with rules ordered by class & coverage facilitating the comparison between the two dataset classes. In this new visualization, it is apparent that low feature values appear to be related to class B whereas higher values to class M (goals **G1**, **Q1**, and **Q2**). In this example, filtering aids in focusing on what is important regarding the overall model behavior, removing unimportant information and reducing cluttering, relying on the so-called Schneiderman's visualization mantra [50].

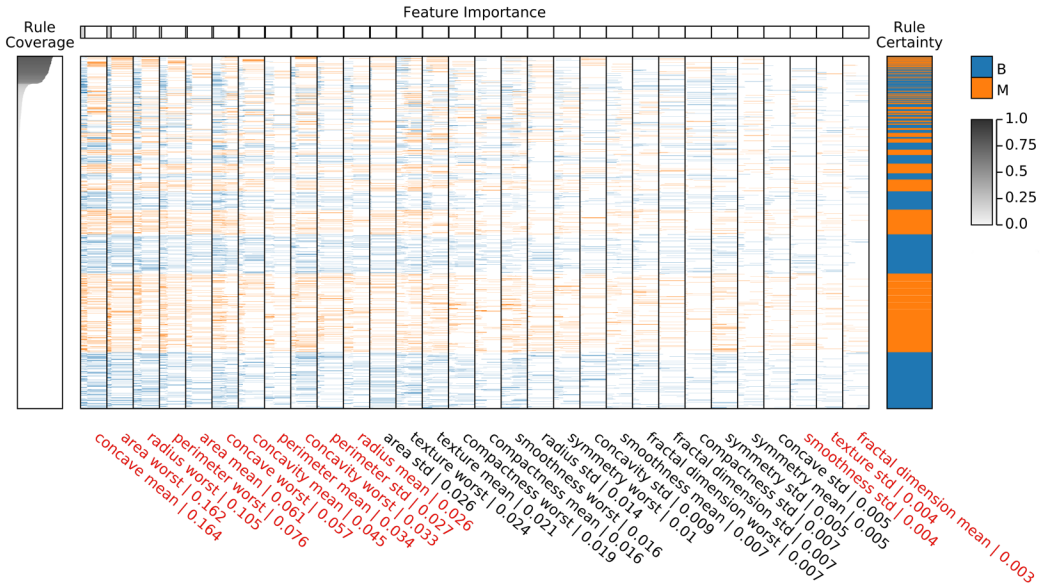
The error rate of 1% in this model is due to the misclassification of only one instance of the test set. Instance x_{29} was wrongly classified as class B with a probability of 55%. Fig. 6(a) shows the ExMatrix LE/UR representation (see Sect. 3.3.2) using x_{29} as target instance. In this visualization, the matrix is ordered by class & coverage to focus on the difference between classes, and some interesting patterns are visible. For instance, predicate ranges of both classes B and M overlap for most features, except for "fractal dimension std" and "concave std". Also, these two features, along with "symmetry std", "concave mean", "compactness std", and "symmetry mean" are more related to class B (blue) since rules of such class heavily use them and sparsely used by rules of class M (orange) showing what is actively used by the model to make the prediction (goals **G2** and **Q3**). Besides, analyzing ExMatrix LE/SC visualization on Fig. 6(b), one can notice that positive changes on features "concave mean" and "perimeter worst" may tie or alter the prediction of x_{29} to class M since many green cells can be observed in the respective columns for rules of class M, while negative changes on "area worst" and "concavity means" increases its classification as class B since many purple cells can be observed in the respective columns for rules of class B (goals **G3** and **Q4**).

4.2 Usage Scenario I: German Credit Bank

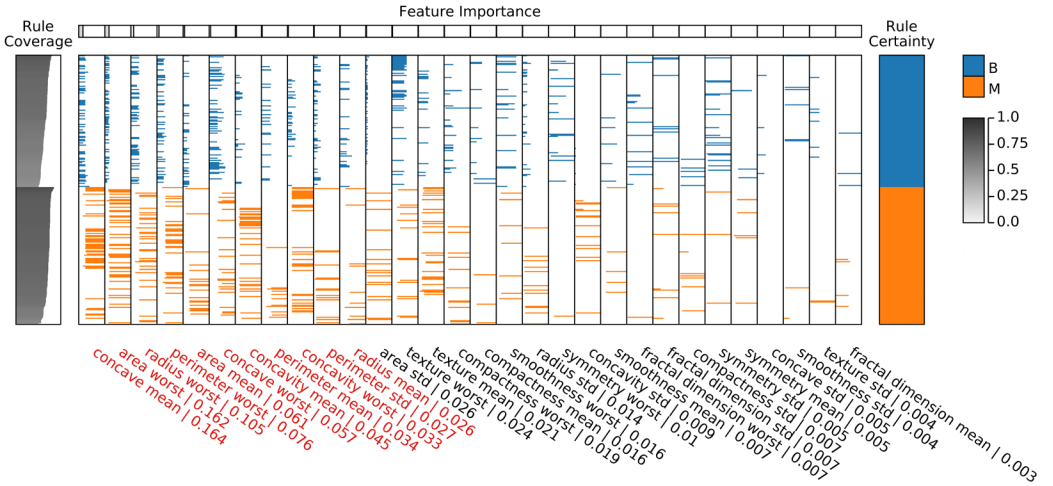
As a first hypothetical usage scenario, we describe a bank manager Sylvia incorporating ExMatrix in her data analytics pipeline. To speed up the evaluation of loan applications, she sends her dataset of years of experience to a data science team and asks for a classification system to aid in the decision-making process. Such dataset contains 1,000 instances (customers profiles) and 9 features (customers information), with 700 customers presenting rejected applications and 300 accepted (here we use a pre-processed [61] version of German Credit Data from UCI). For the implementation of such a system, Sylvia has two main requirements: (1) the system must be precise in classifying loan applications, and; (2) the classification results must be interpretable so she can explain the outcome.

To fulfill the requirements, the data science team builds an RF model setting the number of DTs to 32 with a maximum depth of 6. The

[1] https://popolinneto.gitlab.io/exmatrix/papers/2020/ieeevast/usecase/
[2] https://popolinneto.gitlab.io/exmatrix/papers/2020/ieeevast/usagescenarioi/
[3] https://popolinneto.gitlab.io/exmatrix/papers/2020/ieeevast/usagescenarioii/



(a) ExMatrix GE visualization.

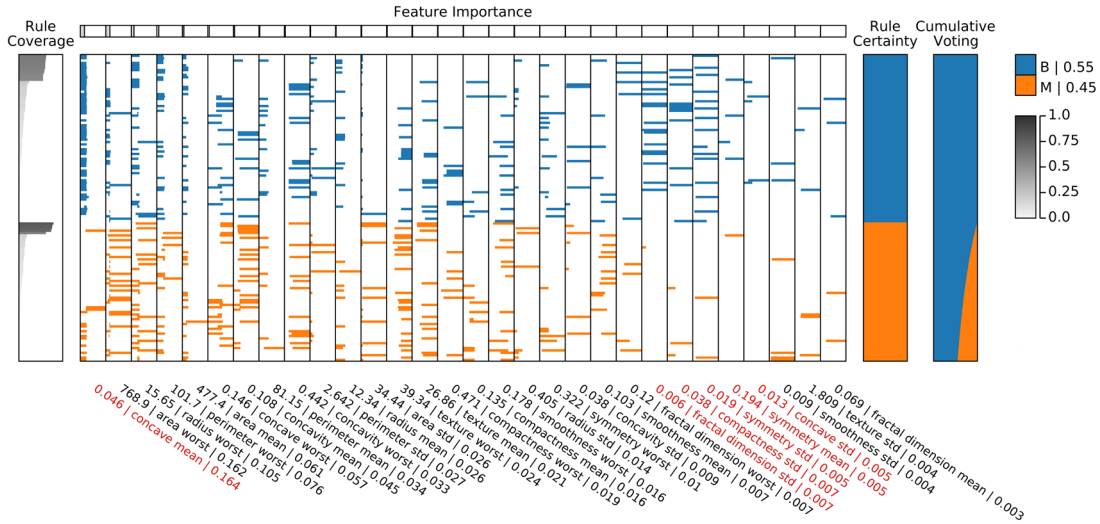


(b) ExMatrix GE representation with filtered rules (only high-coverage rules).

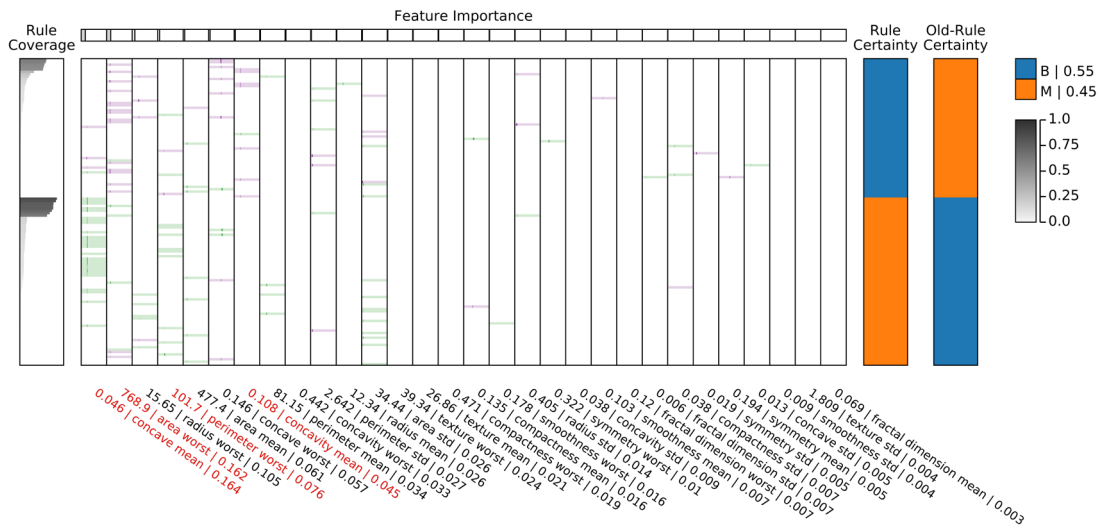
Fig. 5. ExMatrix GE representations of the WDBC RF model. In (a), giving the ordering scheme by rule coverage and feature importance, some patterns emerge in terms of predicates ranges. In (b) the low-coverage rules are filtered-out to help focusing the analysis on what is important. Low feature values appear to be more related to class B whereas higher values to class M for the most important features.

produced model's accuracy was 81%, resulting in 1,273 logic rules, 38.7 rules per DT. Using the ExMatrix GE representation (omitted due to space constraints, see supplemental material), she observes that the features "Account Balance", "Credit Amount", and "Duration of Credit" are the three most important, whereas "Value Savings/Stocks", "Duration in Current address", and "Instalment percent" are the three less. Also, by inspecting the most generic knowledge learned by the system (patterns formed by high-coverage rules) using a filtered representation of the ExMatrix GE visualization on Fig. 7(a), she notices that applications that request a credit to be paid in more extended periods (third column) tend to be rejected, matching her expectations. However, unexpectedly, customers without account ("Account Balance": 1 - No account, 2 - No balance, 3 - Below $200 , 4 - $200 or above) have less chance to have their application rejected (first column), something she did not anticipate (goals **G1**, **Q1**, and **Q2**). Although confronting some of her expectations and bias, she trusts her data, and the classification accuracy seems convincing, so she decides to put the system in practice.

One day she receives a new customer interest in a loan. After filling the system with his data, unfortunately, the application got rejected by the classification system. Based on the new *European General Data Protection Regulation* [11,27,39] that requires explanations about decisions automatically made, the customer requests clarification. By inspecting the ExMatrix LE/UR visualization on Fig. 7(b), she notices, besides the denied probability of 65%, that even if all "approved" rules (blue) are used, very few high-certainty "denied" rules (orange) define the final decision of the model (see the Cumulative Voting and Rule Certainty columns), indicating that those rules, and the related logic statements, have a strong influence in the loan rejection. Also, she sees that the feature "Length of current employment" is the most directly related to the denied outcome since it is used only by rules that result in rejection (goals **G2** and **Q3**). Using this information, she explains to the customer that since he is working for less than one year in the current job (2 as "Length of current employment" corresponds to less than 1 year), the bank recommends denying the application. However, analyzing the ExMatrix LE/SC representation in Fig. 7(b), she realizes that negative changes in features "Credit Amount" and "Duration of Credit" may turn the outcome to approved (goals **G3** and **Q4**). Thereby, as an alternative, she suggests lowering the requested amount and the number of installments. Based on the observable differences to make the rules change class, she notices that upon reducing the credit application from $1,207 to $867 and the number of payments from 24 to 15, the system changes recommendation to "approved". Fig. 7(d)



(a) ExMatrix LE/UR for instance x_{29}, showing the used rules on the classification process.



(b) ExMatrix LE/SC for instance x_{29}, presenting changes in the instance feature values to make the DTs to change class prediction.

Fig. 6. ExMatrix local explanations of the WDBC RF model. Two different visualizations are displayed, one showing the rules employed in the classification of a target instance (a), and one presenting the smallest changes to make the trees of the model to change the prediction of that instance (b). In both cases, the target instance is the only misclassified instance.

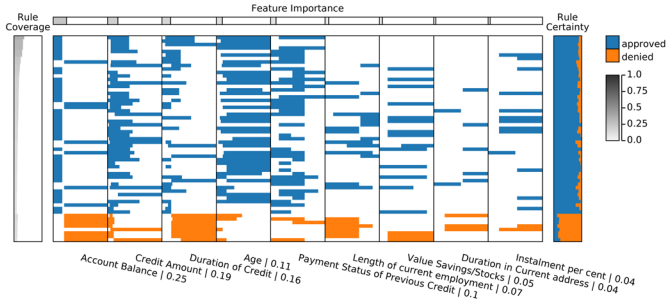
presents the ExMatrix LE/UR visualization if such suggested values are used, changing the final classification.

4.3 Usage Scenario II: Contraceptive Method

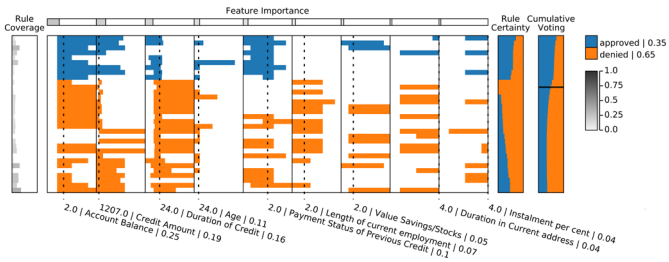
This last usage scenario presents Christine, a public health policy manager who wants to create a contraceptive campaign to advertise a new, safer drug for long term use. To investigate married wives' preferences, Christine's data science team creates a prediction model using a data set with information about contraceptive usage choices her office collected past year (here we use the Contraceptive Method Choice dataset from UCI). The dataset contains $1,473$ samples (married wives profiles) with 9 features, where each instance belongs to one of the classes "No-use", "Long-term", and "Short-term", regarding the contraceptive usage method, with 42.7% of the instances belonging to class No-use, 22.6% to Long-term, and 34.7% to Short-term.

Since interpretability is mandatory in this scenario, allowing the results to be used in practice, the data science team creates an RF model and employs ExMatrix to support analysis. To create the model, the team set the number of DTs to 32 and maximum depth to 6, resulting in $1,383$ logic rules, 43.2 rules per DT. The RF model accuracy is 63%, and, although not ideal for individual classifications, can be used to understand general knowledge learned by the model from the dataset.

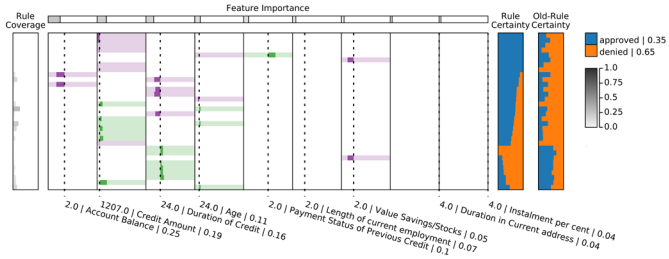
By inspecting the ExMatrix GE representation of the model (omitted due to space constraints, see supplemental material), she readily understands that the features "Number of children ever born", "Wife age", and "Wife education" are the three most relevant for defining the contraceptive method class, while "Media exposure", "Wife now working?", and "Wife religion" are the three less. Also, further exploring a filtered version of the ExMatrix GE representation on Fig. 8, to focus only on high-coverage and high-certainty rules ordered by class, she notices some interesting patterns regarding features space ranges and classes. For instance, lower values for the feature "Number of children ever born" (first column) are more related to class No-use and rarely related to class Long-term. For contraceptive method usage, higher values for the feature "Wife age" (second column) are related to class Long-term, while average and lower values are more related to class Short-term. Also, higher values for "Wife education" (third column) are more related to class Long-term (goals **G1**, **Q1**, and **Q2**). Based on these observations, and given the modest budget she received for the campaign, Christine decides to focus on the group of older and highly educated wives with at least one child to target the campaign's first phase.



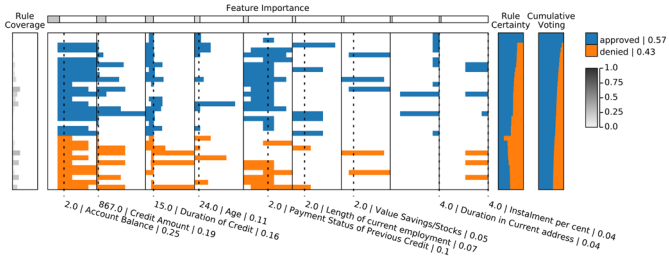
(a) ExMatrix GE showing rules filtered by coverage and certainty.



(b) ExMatrix LE/UR for instance x_{154}.



(c) ExMatrix LE/SC for instance x_{154}.



(d) ExMatrix LE/UR modifying instance x_{154}, which changes RF's decision.

Fig. 7. ExMatrix explanations of a RF model for the German Credit Data UCI dataset. Based on the most generic knowledge learned by the RF model (rules with high coverage) (a), it is possible to conclude that applications requesting credit to be paid in longer periods tend to be rejected. Analyzing one sample (instance x_{154}) of rejected application (c), it is possible to infer that it is probably rejected due to the (applicant) short period working in the current job. However, lowering the requested amount as well as the number of instalments can change the RF's decision (d) and (e).

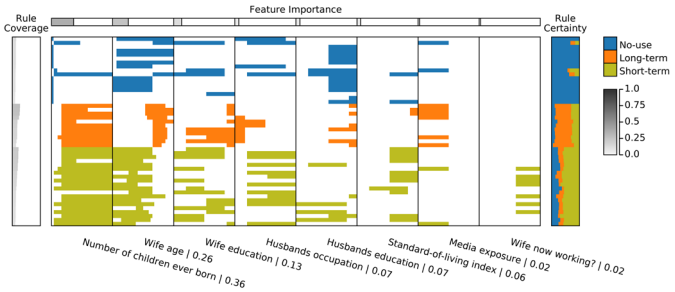


Fig. 8. ExMatrix GE representation (rules filtered by coverage and certainty) of the RF model for the Contraceptive Method Choice UCI dataset. Based on high-coverage high-certainty rules, some interesting patterns can be observed. For instance, on contraceptive method usage, older women tend to use long-term contraceptive methods.

4.4 User Study

To evaluate the ExMatrix method, we performed a user study to assess the proposed visual representations for global and local explanations. In this study, we asked four different questions based on the ExMatrix visualizations created for the use-case presented in Sect. 4.1, focusing on evaluating the goals presented in Table 1.

The study started with video tutorials about RF basic concepts and how to use ExMatrix to analyze RF models and classification results through the proposed explanations. A total of 13 users participated, 69.2% male and 30.8% female, aged between 24 to 36, all with a background in machine learning. The participants were asked to analyze the explanations of Fig. 5(a), Fig. 6(a), and Fig. 6(b), where each analysis was followed by different question(s) (see Table 2). On the visualizations, features names were replaced by "Feature 1" to "Feature 30" and classes names by "Class A" and "Class B", aiming at removing any influence of knowledge domain in the results, since our focus is to assess the visual metaphors.

Using the ExMatrix GE representation (Fig. 5(a)), 76.9% of the participants were able to identify patterns involving feature space ranges and classes, where, for high coverage rules, low features values are more related to class B, while features with large values are more related to class M (**Qst 1**). Using the ExMatrix LE/UR (Fig. 6(a)), also 76.9% of the participants were able to recognize that feature "concave std" is the most related to class B for instance x_{29} classification outcome (**Qst 2**). Using the ExMatrix LE/SC (Fig. 6(b)), 61.5% of the participants were able to identify that negative changes on instance x_{29} features "area worst" and "concavity mean" values would better support the class B outcome (**Qst 3**), and 46.2% were able to identify that positive changes on features "concave mean" and "perimeter worst" values may alter the outcome from class B to class M (**Qst 4**).

In general, the results were promising for the first two analyses, but the participants present worse results when interpreting the ExMatrix LE/SC visualization. This is not surprising since this representation requires a much better background about RF theory. The ExMatrix GE and LE/UR visualizations are more generic and involve much fewer concepts about how RF models work internally. In contrast, the ExMatrix LE/SC requires a good level of knowledge about ensembles models and how the voting system work when making a prediction. Although most of the users self-declared with a background in machine learning, only 30% are RF experts.

We also have asked subjective, open questions, and, in general, users gave positive feedbacks about ExMatrix explanations, where the visualizations were classified as visually pleasing and useful for understanding RF models.

5 Discussion and Limitations

Although the natural choice to visualize a tree collection is to use tree structure metaphors, two main reasons make disjoint rules organized into tables a better option when analyzing DTs and especially RFs. First, using tree structure metaphors, the visual comparison of logic rules (decision paths) can be overwhelming since different paths from the root to the leaves define different orders of attributes, slowing down users when searching within a tree to answer classification questions [22, 29]. An issue that is amplified in RFs, since multiple DTs are analyzed collectively. In contrast, in a matrix metaphor, the attributes are considered in the same order easing this process [22, 29]. Second, given the constraints of usual DT inference methods (non-overlapping predicates with open intervals), features can be used multiple times in a

Table 2. User study questions.

Question	Goals	Visualization
Qst 1 - About features space ranges and class ASSOCIATIONS. Considering rules with HIGH COVERAGE, and features with HIGH IMPORTANCE, select your answer: (three options of associations)	G1, Q1, and Q2	Fig. 5(a)
Qst 2 - Instance 29 is classified as Class A with a probability of 55%, against 45% for Class B. What feature is more related to Class A and less related to Class B? (four options of features)	G2 and Q3	Fig. 6(a)
Qst 3 - Select the pair of features where DELTA CHANGES on instance 29 will potentially INCREASE Class A probability, and by that may SUPPORT its classification as Class A. (four options of features pairs)	G3 and Q4	Fig. 6(b)
Qst 4 - Select the pair of features where DELTA CHANGES on instance 29 will potentially INCREASE Class B probability, and by that may ALTER its classification as Class A. (four options of features pairs)	G3 and Q4	Fig. 6(b)

single decision path resulting in multiple nodes (one per test) using the same feature. Consequently, if tree structures are employed, each feature's decision intervals need to be mentally composed by the user, and nodes using the same feature can be far away in the decision path. The decision intervals are explicit in the matrix representation and can be easily compared across multiples rules and trees. Therefore, although tree structure visual metaphors are the usual choice when hierarchical structures are the focus [25, 48], on DTs and RFs, the decision paths are the object of analysis [22, 29, 37, 54] and transforming paths into disjoint rules organized into tables emphasize what is essential (see supplemental material).

Considering the above points, it is clear that scalability for RFs visualization is not just a choice of getting a visual metaphor that can represent millions of nodes, but getting a visual representation that is scalable and still properly supports essential analytical tasks (see Table 2). Something much more complex than merely visualizing a forest of trees. In this scenario, ExMatrix renders a promising solution, supporting the analysis of many more rules concomitantly than the existing state-of-the-art techniques. However, it is not a perfect solution. ExMatrix covers two different perspectives of RFs, conveying Global and Local information. In the Local visualization, scalability is not a problem since one rule is used per DT, so even for RFs with hundreds or even thousands of trees, ExMatrix scales well. However, for Global visualization, scalability can be an issue since the number of rules substantially grows with the number of trees. Although we can represent one rule per line of pixels, we are limited by the display resolution, and, even when the display space suffices, ExMatrix layouts can be cluttered and tricky to explore.

The solution we adopt to address scalability was to implement the so-called Schneiderman's visualization mantra [50], allowing users to start with an overview of the model, getting details-on-demand by filtering rules to focus on specific sets of interest. Although users are free to select any subset of rules, considering that the goal of the Global visualization is to generate insights about the overall models' behavior, here we mainly explore filtering low-coverage rules since they are only valid for a few specific data instances (that is the coverage definition). Although simple, such a solution makes the analysis of entire models easier by removing unimportant information and reducing cluttering. Another potential solution is to make the rows' height proportional to coverage or certainty so that the rules with the lowest coverage or certainty are less prominent (visible) and could even be combined in less than one line of pixels. We have not tested this approach and left it as future work.

Regarding the user study, although the results were satisfactory and within what we expect for the ExMatrix GE and LE/UR visualizations, the results for the ExMatrix LE/SC representation were sub-optimal, and the *XAI Question Bank* [36] can help us to shed some light about the reasons. According to this bank, the GE addresses the leading question "*How (global)*", whereas the LE/UR addresses the leading question "*Why*", enabling to answer inquiries such as"*What are the top rules/features it uses?*" and "*Why/how is this instance given this prediction?*". However, the LE/SC involves three leading questions, "*What If*", "*How to be that*", and "*How to still be this*", where the changes on instance feature values are presented supporting hypotheses (not answers), which shown to be too complex for the users. We believe that designing visual representations to answer each of these questions individually would be more effective and may reach better results.

Nevertheless, as discussed in the User Study section, participants' low performance not only resulted from the visual metaphor but also the expertise on RF models. Among the participants, few know the RF technique in detail, indicating that people with less expertise can use ExMatrix GE and LE/UR visualizations, but the LE/SC representation is more suitable for experts. In general, despite the complexity of the questions we ask participants to solve, they acknowledged the ExMatrix potential, expressing encouraging remarks, including "*... this solution ... allows a deeper understanding of how each particular rule or feature impacted on the final the decision/classification.*" or "*I think the ExMatrix can be used in a variety of domains, from E-commerce to Healthcare...*".

Although we design ExMatrix with RF interpretability in mind, it can be readily applied to DT models, such as the ones used as surrogates for black-box models as Artificial Neural Networks and Support Vector Machines, or approaches based on logic rules such as Decision Tables since the core of our method is the visualization of rules. Another compelling scenario that can be explored is the engineering of models. In this case, through rule selection and filtering, simplified models could be derived where, for instance, only high coverage rules are employed or any other subset of interest. Also, model construction and improvement can be supported. The visual metaphors we propose can be easily applied to the analysis and comparison of RF models resulting from different parametrizations, such as different numbers of trees and their maximum depth. Therefore, allowing machine learning engineers to go beyond accuracy and error when building a model.

6 Conclusions and Future Work

In this paper, we present *Explainable Matrix (ExMatrix)*, a novel method for Random Forest (RF) model interpretability. ExMatrix uses a matrix-like visual metaphor, where logic rules are rows, features are columns, and rules predicates are cells, allowing to obtain overviews of models (Global Explanations) and audit classification results (Local Explanations). Although simple, ExMatrix visual representations are powerful and support the execution of tasks that are challenging to perform without proper visualizations. To attest ExMatrix usefulness, we present one use-case and two hypothetical usage scenarios, showing that RF models can be interpreted beyond what is granted by usual metrics, like accuracy or error rate. Although our primary goal is to aid in RF models global and local interpretability, the ExMatrix method can also be applied for the analysis of Decision Trees, such as the ones used as surrogates models, or any other technique based on logic rules, opening up new possibilities for future development and use. We plan as future work to create new ordering and filtering criteria along with aggregation approaches to improve the current ExMatrix explanations and, more importantly, to conceive new ones. Another fascinating forthcoming work is creating optimized rule-based models from complex RF models, which we also intend to investigate.

Acknowledgments

The authors wish to thank the valuable comments and suggestions obtained from the reviewers, as well as the support received from the Qualification Program of the Federal Institute of São Paulo (IFSP). We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] B. Alper, B. Bach, N. Henry Riche, T. Isenberg, and J.-D. Fekete. Weighted graph comparison techniques for brain connectivity analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI 13, page 483492, New York, NY, USA, 2013. Association for Computing Machinery.

[2] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. Visualizing Sets and Set-typed Data: State-of-the-Art and Future Challenges. In R. Borgo, R. Maciejewski, and I. Viola, editors, *EuroVis - STARs*. The Eurographics Association, 2014.

[3] M. Ankerst, C. Elsen, M. Ester, and H.-P. Kriegel. Visual classification: An interactive approach to decision tree construction. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 392–396, New York, NY, USA, 1999. ACM.

[4] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, and J.-D. Fekete. Matrix reordering methods for table and network visualization. *Computer Graphics Forum*, 35(3):693–716, 2016.

[5] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, and J.-D. Fekete. Matrix reordering methods for table and network visualization. *Computer Graphics Forum*, 35(3):693–716, 2016.

[6] G. Biau and E. Scornet. A random forest guided tour. *TEST*, 25(2):197–227, Jun 2016.

[7] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[8] L. Breiman. Manual on setting up, using, and understanding random forests v3. 1. *Statistics Department University of California Berkeley, CA, USA*, 1:58, 2002.

[9] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[10] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.

[11] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8), 2019.

[12] C.-H. Chen, H.-G. Hwu, W.-J. Jang, C.-H. Kao, Y.-J. Tien, S. Tzeng, and H.-M. Wu. Matrix visualization and information mining. In J. Antoch, editor, *COMPSTAT 2004 — Proceedings in Computational Statistics*, pages 85–100, Heidelberg, 2004. Physica-Verlag HD.

[13] C.-h. Chen, A. Sinica, and Taipei. Generalized association plots: information visualization via iteratively generated correlation matrices. *Statistica Sinica*, 12:7–29, 01 2002.

[14] J. Choo, H. Lee, J. Kihm, and H. Park. ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pages 27–34, Oct 2010.

[15] J. A. Cruz and D. S. Wishart. Applications of machine learning in cancer prediction and prognosis. *Cancer Informatics*, 2:117693510600200030, 2006.

[16] D. Dheeru and E. Karra Taniskidou. UCI machine learning repository, 2017.

[17] F. Di Castro and E. Bertini. Surrogate decision tree visualization interpreting and visualizing black-box classification models with surrogate decision tree. *CEUR Workshop Proceedings*, 2327, 1 2019.

[18] T.-N. Do. Towards simple, easy to understand, an interactive decision tree algorithm. *College Inf. Technol., Can tho Univ., Can Tho, Vietnam, Tech. Rep*, pages 06–01, 2007.

[19] M. Du, N. Liu, and X. Hu. Techniques for interpretable machine learning, 2018.

[20] A. Endert, W. Ribarsky, C. Turkay, B. W. Wong, I. Nabney, I. D. Blanco, and F. Rossi. The state of the art in integrating machine learning into visual analytics. *Computer Graphics Forum*, 36(8):458–486, 2017.

[21] R. A. FISHER. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[22] A. A. Freitas. Comprehensible classification models: A position paper. *SIGKDD Explor. Newsl.*, 15(1):1–10, Mar. 2014.

[23] T. Fujiwara, O. Kwon, and K. Ma. Supporting analysis of dimensionality reduction results with contrastive learning. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):45–55, Jan 2020.

[24] O. Gomez, S. Holter, J. Yuan, and E. Bertini. Vice: Visual counterfactual explanations for machine learning models. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI 20, page 531535, New York, NY, USA, 2020. Association for Computing Machinery.

[25] M. Graham and J. Kennedy. A survey of multiple tree visualisation. *Information Visualization*, 9(4):235252, Dec. 2010.

[26] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti. Local rule-based explanations of black box decision systems, 2018.

[27] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5):93:1–93:42, Aug. 2018.

[28] P. Hall. On the art and science of machine learning explanations, 2018.

[29] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141 – 154, 2011.

[30] B. Hferlin, R. Netzel, M. Hferlin, D. Weiskopf, and G. Heidemann. Interactive learning of ad-hoc classifiers for video visual analytics. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 23–32, Oct 2012.

[31] R. Kohavi. The power of decision tables. In *Proceedings of the 8th European Conference on Machine Learning*, ECML'95, pages 174–189, Berlin, Heidelberg, 1995. Springer-Verlag.

[32] J. Krause, A. Dasgupta, J. Swartz, Y. Aphinyanaphongs, and E. Bertini. A workflow for visual diagnostics of binary classifiers using instance-level explanations. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 162–172, Oct 2017.

[33] H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1675–1684, New York, NY, USA, 2016. ACM.

[34] T. Lee, J. Johnson, and S. Cheng. An interactive machine learning framework. *CoRR*, abs/1610.05463, 2016.

[35] J. Lei, Z. Wang, Z. Feng, M. Song, and J. Bu. Understanding the prediction process of deep networks by forests. In *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, pages 1–7, Sep. 2018.

[36] Q. V. Liao, D. Gruen, and S. Miller. Questioning the ai: Informing design practices for explainable ai user experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI 20, page 115, New York, NY, USA, 2020. Association for Computing Machinery.

[37] E. Lima, C. Mues, and B. Baesens. Domain knowledge integration in data mining using decision tables: case studies in churn prediction. *Journal of the Operational Research Society*, 60(8):1096–1106, 2009.

[38] S. Liu, J. Xiao, J. Liu, X. Wang, J. Wu, and J. Zhu. Visual diagnosis of tree boosting methods. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):163–173, Jan 2018.

[39] X. Liu, X. Wang, and S. Matwin. Interpretable deep convolutional neural networks via meta-learning. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, July 2018.

[40] W.-Y. Loh. Fifty years of classification and regression trees. *International Statistical Review*, 82(3):329–348, 2014.

[41] M. Migut and M. Worring. Visual exploration of classification models for risk assessment. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pages 11–18, Oct 2010.

[42] Y. Ming, H. Qu, and E. Bertini. Rulematrix: Visualizing and understanding classifiers with rules. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):342–352, Jan 2019.

[43] J. G. S. Paiva, W. R. Schwartz, H. Pedrini, and R. Minghim. An approach to supporting incremental visual data classification. *IEEE Transactions on Visualization and Computer Graphics*, 21(1):4–17, Jan 2015.

[44] P. E. Rauber, S. G. Fadel, A. X. Falco, and A. C. Telea. Visualizing the hidden activity of artificial neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):101–110, Jan 2017.

[45] 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*, KDD 16, page 11351144, New York, NY, USA, 2016. Association for Computing Machinery.

[46] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. 2018.

[47] H. Schulz. Treevis.net: A tree visualization reference. *IEEE Computer Graphics and Applications*, 31(6):11–15, Nov 2011.

[48] H. Schulz, S. Hadlak, and H. Schumann. The design space of implicit hierarchy visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):393–411, 2011.

[49] S. seok Choi and S. hyuk Cha. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, pages 43–48, 2010.

[50] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[51] G. Stiglic, M. Mertik, V. Podgorelec, and P. Kokol. Using visual interpretation of small ensembles in microarray analysis. In *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*, pages 691–695, June 2006.

[52] E. Strumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.*, 11:1–18, Mar. 2010.

[53] J. Talbot, B. Lee, A. Kapoor, and D. S. Tan. Ensemblematrix: Interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1283–1292, New York, NY, USA, 2009. ACM.

[54] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. Pearson, 1 edition, 2005.

[55] S. T. Teoh and K.-L. Ma. Paintingclass: Interactive construction, visualization and exploration of decision trees. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 667–672, New York, NY, USA, 2003. ACM.

[56] S. Tzeng, H. Wu, and C. Chen. Selection of proximity measures for matrix visualization of binary data. In *2009 2nd International Conference on Biomedical Engineering and Informatics*, pages 1–9, Oct 2009.

[57] S. van den Elzen and J. J. van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 151–160, Oct 2011.

[58] H.-M. Wu, S. Tzeng, and C.-h. Chen. *Matrix Visualization*, pages 681–708. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[59] M. Wu, M. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. 2018.

[60] F. Yang, M. Du, and X. Hu. Evaluating explanation without ground truth in interpretable machine learning. *CoRR*, abs/1907.06831, 2019.

[61] X. Zhao, Y. Wu, D. L. Lee, and W. Cui. iforest: Interpreting random forests via visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):407–416, Jan 2019.