TreePOD: Sensitivity-Aware Selection of Pareto-Optimal Decision Trees

Thomas Mühlbacher, Lorenz Linhardt, Torsten Möller, and Harald Piringer

Abstract—Balancing accuracy gains with other objectives such as interpretability is a key challenge when building decision trees. However, this process is difficult to automate because it involves know-how about the domain as well as the purpose of the model. This paper presents TreePOD, a new approach for sensitivity-aware model selection along trade-offs. TreePOD is based on exploring a large set of candidate trees generated by sampling the parameters of tree construction algorithms. Based on this set, visualizations of quantitative and qualitative tree aspects provide a comprehensive overview of possible tree characteristics. Along trade-offs between two objectives, TreePOD provides efficient selection guidance by focusing on Pareto-optimal tree candidates. TreePOD also conveys the sensitivities of tree characteristics on variations of selected parameters by extending the tree generation process with a full-factorial sampling. We demonstrate how TreePOD supports a variety of tasks involved in decision tree selection and describe its integration in a holistic workflow for building and selecting decision trees. For evaluation, we illustrate a case study for predicting critical power grid states, and we report qualitative feedback from domain experts in the energy sector. This feedback suggests that TreePOD enables users with and without statistical background a confident and efficient identification of suitable decision trees.

Index Terms—Model selection, classification trees, visual parameter search, sensitivity analysis, Pareto optimality

1 Introduction

Decision trees are a common technique for statistical classification. Hierarchical decision rules model *classes* of a categorical variable depending on numerical or categorical independent variables, called *features*. The decision rules are typically inferred from *training data* for which the classes are known, which is referred to as supervised learning [14]. Frequent types of rules include thresholds on numerical features and class membership vectors on categorical features. In contrast to other types of classification models such as neural networks, a key advantage of decision trees is the ability of humans to understand how the model works. Experts in many fields such as medical diagnosis, image processing, or fraud detection therefore appreciate decision trees for their *interpretability* [14, 19]. In addition to classifying new data instances, the understandable model structure also supports explaining class dependencies for hypothesis generation and reporting.

The process of building decision trees involves multiple trade-offs. As for other model types, the most well-known trade-off is that between over- and underfitting the data for robust generalization (bias-variance trade-off). Automated techniques exist which adjust the model complexity accordingly, e.g., by using different data for growing and pruning the tree [14]. In addition to *accuracy*, however, aspects regarding model *interpretability* by humans are often equally important for decision trees. Model interpretability has received much attention recently [12, 15, 19] and is a multi-faceted goal by itself. Simple trees with limited depth and comprising only few decision rules based on a small number of features are typically easier to understand. Moreover, decision trees intended for human decision makers benefit from nice, round thresholds [15] (e.g., $x \leq 100$ instead of $x \leq 99.475$).

Balancing accuracy gains, interpretability and other objectives such as feature acquisition costs [10, 22, 45] is a key challenge when building decision trees. However, this process is difficult to automate because it involves know-how about the domain as well as the purpose of the model and often requires a qualitative assessment of the decision tree by domain experts. Even with a deep understanding of the learning algorithm, obtaining a decision tree that satisfies all objectives takes substantial time for trial-and-error [32]. Aggravating the challenge, many domain experts do not have a background in statistical learning [46], but still need to build decision trees which meet their objectives while reflecting their domain knowledge.

This paper proposes TreePOD, a new Visual Analytics technique for decision tree identification which addresses these challenges. Inspired by work on visual parameter space exploration [40] and in line with recent work in statistics [49] , our approach is based on exploring a large set of tree candidates. A key goal is to support a *global-to-local strategy for model selection* (G1) that initially provides the user with a comprehensive overview of possible tree characteristics. A second goal is to *address users with and without deep statistical background* (G2). For this reason, TreePOD takes a result-oriented approach which focuses on characteristics of generated trees such as prediction accuracy, complexity, and interpretability. Details of the machine learning process (e.g., training parameters) are hidden by default and exposed only at request. In order to foster a *quick identification of suitable trees* (G3), TreePOD supports an effective quantitative and qualitative comparison of model alternatives. In order to further *increase the user confidence in the selected model* (G4), TreePOD visualizes the sensitivity of tree candidates on variations of generation parameters.
Based on TreePOD as the main contribution of this paper, additional contributions include:

- An outlined workflow for decision tree selection.
- A case study to address a real-world problem in the energy sector.
- Qualitative feedback of domain experts from the energy sector.

2 Related Work

Research in statistical learning has devised many automated algorithms for building decision trees, e.g., CART [6], C4.5 [37], and CHAID [16]. Many of these algorithms use entropy minimization to choose features and split positions when growing the tree. After the growing phase, automated approaches can be used to ensure the generalizability of the model, e.g., by pruning and cross validation [14]. Decision trees have also been extended to ensemble learning techniques such as random forests. Such approaches may further increase the accuracy at the cost of incurring significantly higher complexity compared to single trees. Gleicher [12] notes that accuracy is not the only concern and mentions efficiency, generalizability, robustness, conciseness, verifiability, self-consistency, and comprehensibility as some other qualities that model designers must consider. Gleicher also stresses that these properties form trade-offs where the

- *Thomas Mühlbacher and Harald Piringer are with the VRVis Research Center. Email: {tm | hp}@vrvis.at*
- *Lorenz Linhardt is with ETH Zurich. Email: llorenz@student.ethz.ch*
- *Torsten Möller is with the University of Vienna. Email: torsten.moeller@univie.ac.at*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

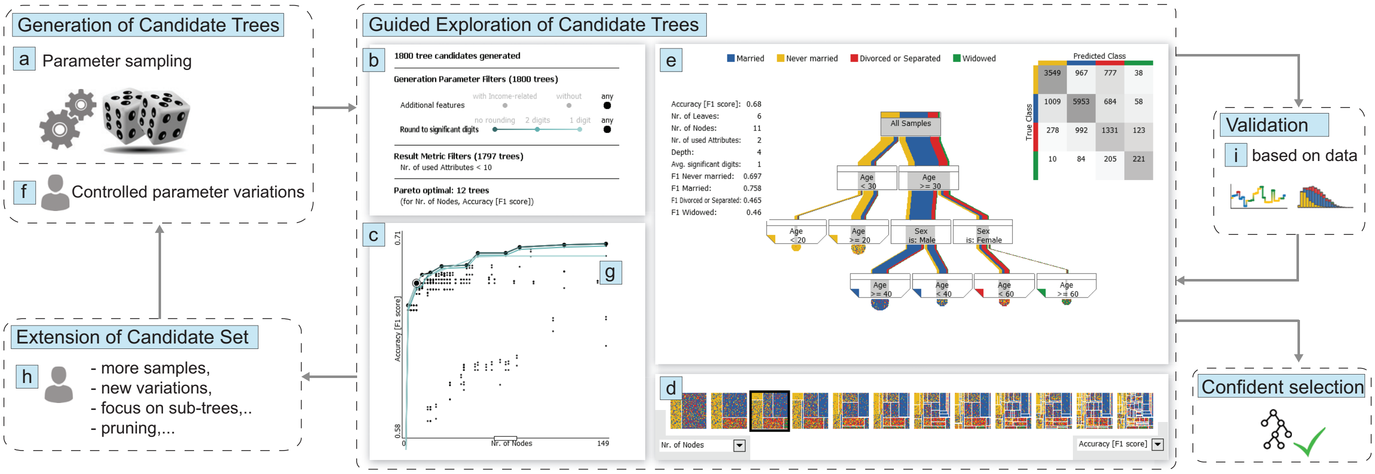


Figure 1. Selection of decision trees explaining *marital status* in the UCI Census Income 1994 dataset [21]. (a) Candidate trees are generated by sampling the parameters of decision tree algorithms. Linked visualizations guide the selection from this set by providing (b) a summary of tree candidates and parameter variations, (c) a sensitivity-aware overview of the trade-off between the conflicting objectives *accuracy* and *number of nodes*, (d) a qualitative comparison of Pareto-optimal trees, and (e) details of a selected decision tree. (f) Applying controlled parameter variations to every tree conveys the effect of parameter changes on tree characteristics, e.g., how rounding of decision boundaries affects accuracy (g). Users can extend the set of candidate trees at any time, (h) and validate trees based on data using linked views.

proper balance depends on the context and needs.

An increasing number of automated approaches take comprehensibility into account as an important goal. Jung et al. [15], for example, perform rounding of model coefficients in logistic regression classifiers in order to make them easier for humans to interpret. Lakkarju et al. [19] include metrics for interpretability in the objective function for model selection. In many cases, however, assessing comprehensibility requires a qualitative inspection by domain experts.

In contrast to such automated approaches, visualization research has focused on cooperative approaches for decision tree construction which enable users to incorporate their domain knowledge in the generation process. Ankerst et al. [3] let the user evaluate intermediate results of the construction algorithm to specify constraints. This enables the computer to automatically create patterns satisfying these constraints. Van den Elzen and van Wijk [46] support an iterative refinement of a tree during the growing, optimization, and pruning phases. This process is based on BaobabView, a technique for visualizing decision trees which combines advantages of other methods such as node-link diagrams [13, 48] and icicle plots [3, 23]. All these cooperative approaches may improve comprehensibility and user confidence in the model. A study by Liu and Salvendy [23] shows that resulting trees have relatively high classification accuracies and small sizes. However, focusing on the iterative refinement of single trees may not lead to the global optimum. Moreover, such approaches do not communicate the overall achieveability of modeling objectives and may require statistical know-how and significant time by the user.

In order to provide a global coverage of possible tree characteristics, some automated approaches obtain multiple decision trees as result. Zhao [50] creates Pareto optimal decision trees to capture the trade-off between different types of misclassification errors. Likewise, Czajkowski and Kretowski [9] use an evolutionary algorithm to generate multiple decision trees which are Pareto optimal for contradictory metrics such as accuracy and the number of nodes. These approaches focus on generating an appropriate set of decision trees, not on exploring this set to facilitate the model selection by a human expert. Czajkowski and Kretowski stress that the comprehensibility of the generated Pareto front is a main issue for future work.

In visualization, an increasing number of systems provide global exploration strategies of parameter spaces [40], e.g., in simulation [1, 7, 25, 35] and image analysis [43]. In many cases, the goal is to identify input parameter values which optimize the output in some sense. Assessing the output often involves both quantitative metrics and qualitative judgments of complex results, for example segmented image data [43]. Statistical model selection is a closely related problem. Understanding the relation between abstract generation parameters and the resulting model is typically non-intuitive and model selection is usually based on multiple quantitative and qualitative criteria. Related work for exploring model spaces include subspace clustering [28], neural networks [26], and association rules [8].

In the context of decision trees, we regard the work by Padua et al. [32] as most similar. Their system supports the analysis of a large set of candidate trees generated by sampling the parameter space of decision tree algorithms. Linked views visualize this parameter space as well as metrics of the resulting trees and thus enable to relate inputs to outputs by interaction. The trees are shown as node-link diagrams and small icicle plots that convey the structure but not the accuracy. This system provides a global overview of tree characteristics (G1) and guides statistical experts towards useful training parameters. However, their work does not explicitly recognize trade-offs between objectives (G3) and does not visualize their sensitivity on changes of generation parameters or the evaluation data.The analysis focuses on an existing set of trees and does not address the integration in an interactive workflow for decision tree building. Moreover, by exposing many details about generation parameters, the system is primarily designed for users with statistical background which contradicts goal (G2).

3 Overview of TreePOD

TreePOD is a new Visual Analytics technique for sensitivity-aware model selection. The key idea is to create a large set of candidate trees that can be explored with respect to objectives such as prediction accuracy, or interpretability. To this end, the parameter space of tree construction algorithms is sampled to create a diverse set of trees (Fig. 1a, Sec. 4). Visualizing the candidate set at different levels of detail in multiple coordinated views [39] enables a global-to-local strategy for model selection [40] (Sec. 5): A summary panel displays a concise description of the candidate set, and provides various ways of focusing on candidate subsets (Fig. 1b). A quantitative overview shows achievable values for pairs of objectives, and guides selection along tradeoffs by identifying the Pareto front, i.e., the set of Pareto-optimal trees (Fig. 1c). Tree maps at the bottom visualize accuracy and complexity of the Pareto-optimal trees in a compact form (Fig. 1d). A detail panel shows the currently selected tree and its characteristics (Fig. 1e).

To investigate local sensitivities of tree characteristics to parameter changes, users can specify a controlled variation of parameters (Fig. 1f, Sec. 6). Visualizing these variations shows how characteristics of single trees, multiple trees, or entire Pareto-fronts are affected by constraints such as rounded decision rules (Fig. 1g). Section 6.3 describes how this approach to *sensitivity-aware trade-off exploration* supports a variety of model selection tasks.

While TreePOD focuses on analyzing and choosing from an existing set of candidates, we also outline its integration in a workflow for

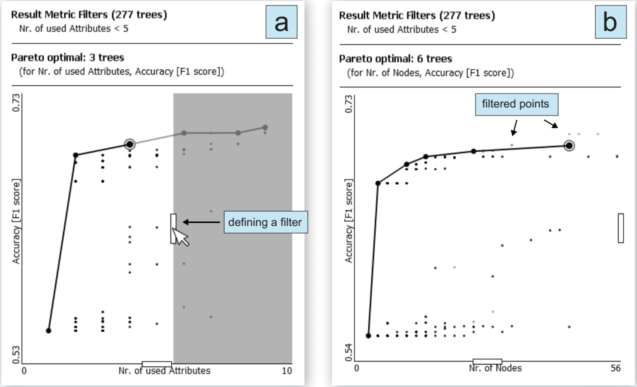

Result Metric Filters (277 trees)
Nr. of used Attributes < 5
a
Pareto optimal: 3 trees
(for Nr. of used Attributes, Accuracy [F1 score])
0.73
Accuracy [F1 score]
0.53
defining a filter
0
Nr. of used Attributes
10
Result Metric Filters (277 trees)
Nr. of used Attributes < 5
b
Pareto optimal: 6 trees
(for Nr. of Nodes, Accuracy [F1 score])
0.73
filtered points
Accuracy [F1 score]
0.54
0
Nr. of Nodes
56

Figure 2. The Pareto front guides tree selection along trade-offs between two result metrics, in this example (a) *accuracy* vs. *nr. of used attributes*, and (b) *accuracy* vs. *nr. of nodes*. Hard constraints on metrics filter the set of tree candidates in all views.

building decision trees (Sec. 7). Key steps in this workflow include the incremental extension of the candidate set based on insights from exploration (Fig. 1h), and the validation of trees based on data (Fig. 1i).

As a guiding example illustrating TreePOD, consider the following fictional scenario: *Jane, an analyst working for the ministry of social affairs, aims to predict the multi-class attribute Marital Status in the UCI "Census Income Dataset 1994" [21]. Features comprise 12 demographic attributes like Age, Sex, Income, Occupation, Native Country, and many others*[1]. *Her goal is to obtain an accurate and concise set of rules suitable for reporting or policy-making.*

4 Generation of candidate trees

A prerequisite for model selection is the availability of good candidate models. Automatic decision tree algorithms help to identify tree candidates efficiently. Based on a specification of various parameters, they produce a decision tree for pre-classified data by heuristic optimization in two distinct phases:

1) Training: Given a subset of training data and training parameters, the algorithm generates an initial tree description. Training parameters include a set of candidate features and a selection criterion that defines a feature selection strategy (e.g., maximizing information gain [37], Gini impurity [6] or gain ratio [38]). Other parameters include numerical termination criteria for the build process such as a maximal tree depth or a minimal leaf size needed for further splits.

2) Post-processing: In the optional second phase, post-processing such as pruning to avoid overfitting [14], or rounding of numerical decision borders to increase interpretability [42, 15] may be applied.

Training and post-processing involve numerical, categorical and set-typed parameters. For easier readability, we subsequently use *parameter value* as an umbrella term for all types of parameters. Choosing parameter values that result in desirable trees is non-trivial and typically requires substantial effort [32]. Instead of forcing the parameter space upon the user, TreePOD constructs a diverse set of candidates by sampling various parameters in a stochastic or pseudo-random fashion. This may include drawing feature subsets, drawing the maximal tree depth from a range (e.g., [1,..,10]), or randomly choosing a tree pruning method. As a key benefit, stochastic assignment of parameters helps creating diverse and unbiased candidates, which increases the probability of reaching the global optimum during exploration. It also reduces the need to specify parameter values prior to exploration.

Users can also manually assign parameters to incorporate knowledge about algorithms [27] or previously obtained insights. This includes setting parameters to a fixed value for all trees (e.g., max depth = 6), as well as manual adjustment of sampling ranges (e.g., max depth ∈ [1,..,6]). However, we provide reasonable defaults for all sets and ranges to keep the mandatory user input to a minimum. Data subsets for growing, pruning, and evaluation can also be manually specified, but are otherwise automatically determined by splitting the available data into random parts of equal size.

TreePOD also supports various common pruning techniques [14]. As the simplest method, we support collapsing sub-trees if all leaves within produce the same classification. Pruning can also be deactivated to allow for a more detailed analysis of achievable accuracy.

In the guiding example, Jane wants to know how well small models can perform. She generates 300 decision trees by sampling (1) the maximal tree depth between 1 and 6, (2) the minimal leaf size required for further splits, (3) as well as subsets of the 12 available features to obtain different explanations. This generates 300 candidates that are evaluated for an exploration of their results (see Figure 2).

5 Guided Exploration of Pareto-Optimal Trees

This section describes interactive visualizations of the tree candidates at different levels of detail. The goal is to support the selection of suitable trees based on quantitative and qualitative characteristics.

5.1 Candidate summary panel

At the coarsest level, TreePOD provides a concise summary of all tree candidates (see Fig. 1b). This view describes how the set of candidates is successively refined by the user during exploration. Users may define *generation parameter* filters, for example to focus on trees based on particular feature subsets or rounding thresholds. Tree candidates may also be filtered based on their *result metrics* such as accuracy (see Section 5.2). The current set of filters is summarized in this view. Furthermore, the panel states the number of *Pareto-optimal* trees regarding two objective metrics, which is used as central guidance concept in TreePOD. These concepts will be introduced in the following sections.

5.2 Quantitative trade-off overview

The model selection process typically involves quantitative metrics. The metrics in our implementation refer to three types of objectives:

(1) Accuracy, as measured by the F1 score (aka F-measure) [51]. We provide per-class scores (e.g., *F1 "Married"*) as well as the overall score by computing the weighted average of F1 across classes (denoted *Accuracy [F1 score]*).

(2) Complexity, optionally expressed as either the total *number of nodes*, the *number of leaves*, the maximum tree *depth*, the *number of used attributes*, or the total *feature cost*.

(3) Interpretability in terms of human-friendly numbers, computed as the *average num. of significant digits* in numerical rules [31].

We do not intend to make a case for any particular metric. The concepts of TreePOD could be applied to other metrics as well.

For an effective quantitative overview of the tree candidates, TreePOD displays two user-specified metrics in a 2D scatter plot (e.g., *Accuracy* vs. *Nr. of used attributes* in Fig. 2a). This provides an overview of the candidates in terms of quantitative characteristics and may reveal patterns such as discontinuities or clusters caused by distinct parameter settings, e.g., the inclusion of important features.

Not all candidates are equally relevant for model selection. For example, among all trees of the same size in Fig. 2b, some are substantially more accurate than others. An established concept in multi-criteria decision making is Pareto optimality [18]. In general, a solution is considered Pareto-optimal if no other solution exists that is better for some criteria without being worse for others. The set of all Pareto optimal solutions is called *Pareto front*. In our case, this front comprises all candidate models which are Pareto optimal regarding the two objectives mapped to the axes of the scatter plot.

Pareto-optimal candidates are highlighted using an increased point size and connected with a line to visualize the Pareto front (see Fig. 2). Drawing the front as an interpolated line rather than step-wise is a potentially too optimistic approximation of the real Pareto front. However, we decided to tolerate this as the selection relies on the discrete set of candidates rather than on the continuous shape of the Pareto front. Visually, drawing interpolated lines enables to compare slopes across neighbouring segments. Very steep and very shallow segments indicate transitions that provide high gain of one objective for low additional cost of the other, guiding users towards possible "sweet spots".

[1] For better demonstration, we intentionally exclude the highly correlated feature *Relationship Status*, as this would yield trivial rules like *Marital Status is 'Married'* if *Relationship Status is: Wife*

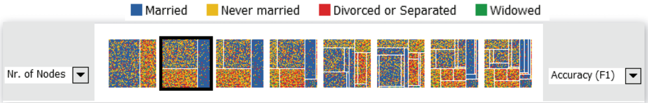


Figure 3. Pixel-based treemaps convey qualitative aspects of accuracy and complexity along a Pareto front.

Any tree can be selected by a click, making it *focal*. In the scatter plot, this *focal tree* is highlighted by a black circle around the point (Fig. 2). Linked views focus on it as well, for example, to show the tree description and parameters which led to that result (see Sec. 5.4).

The view also enables to define range filters for objective values by dragging handles inwards from the plot borders. In Fig. 2, all trees using more than 5 attributes are excluded as indicated by a semi-transparent gray area. Filters persist when changing objectives, which allows investigating a filtered set of candidates with respect to other objectives. This supports a global-to-local workflow for model selection, where the considered set of trees is iteratively refined (G1). Filtered points are not considered when computing the Pareto fronts, but are still displayed in a lower intensity as context. Additionally, a textual representation is shown in the candidate summary (see Fig. 2).

5.3 Qualitative comparison along the Pareto front

The quantitative overview described in the previous section provides effective guidance to trees with high objective values. However, summary metrics hide multiple sources of ambiguity that may be relevant to the decision maker. For example, a high overall accuracy of models can be the result of well-explaining features, or of highly skewed base rates [51]. Likewise, a single accuracy metric does not inform about the distribution of accuracy among the classes.

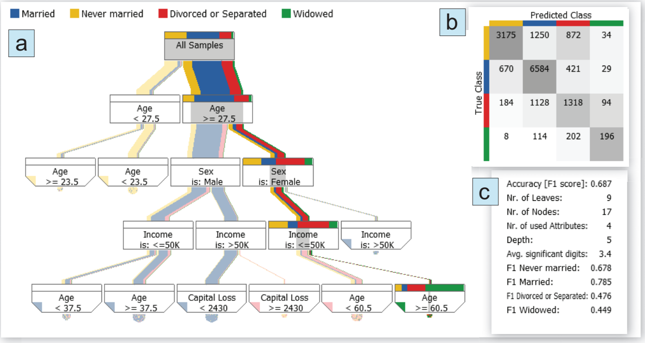
To visualize such qualitative aspects along a trade-off, we encode the set of Pareto-optimal candidates using small *tree maps* [41] (see Fig. 3). Their sequence represents a linear traversal of the 2D Pareto front, i.e. one objective improves while the other deteriorates from left to right. This arrangement facilitates switching to the next more accurate or next simpler Pareto-optimal tree for an efficient browsing of candidates. Clicking a plot makes the corresponding tree *focal*.

Each partition in a tree-map corresponds to a leaf node, with a relative size proportional to the percentage of data instances classified by that leaf. This enables an effective perception of complexity for the corresponding decision tree (see Fig. 3).

Inspired by perception-based approaches to classification [2, 3, 20], we encode the class distribution within a leaf by a quasi-random placement of pixels according to the class frequencies. The emerging pattern enables an intuitive perception of purity and, for high-purity leaves, easy identification of the predominant class. The selected plot in Figure 3, for example, indicates a first split that isolates *Married* persons very well (mostly blue leaf). The other leaves are much less pure. Discriminability of hue depends on the size of coherent areas [29] and thus on the separability of a data set. We found that, in practice, 5-7 classes can be effectively discriminated also for small pure leaves. For noisy leaves, discrimination of single pixels is typically less important than the overall perception of entropy, which is directly supported by the encoding. This encoding has the advantage that both over- and underfitted trees result in high-frequency patterns. Simple and accurate trees, however, contain large, homogeneous regions. This provides effective qualitative guidance along the trade-off.

Our approach to pixel-based encoding of class distribution is inspired by work of Ankerst et al. [3], but differs with respect to two major aspects: first, their approach shows all levels of the tree next to each other, visualizing the purity gained by every split. Our approach focuses on the leaves to enable an efficient comparison of accuracy and complexity across multiple trees. Second, their pixel arrangement is spatially linked to data items. Our pixel placement is random, which avoids visual structure within the leaves that distracts from the perception of tree complexity. Details on the topology and splits of the tree are shown in a linked visualization (see Sec. 5.4).

Inspecting the tree maps in Fig. 1d, Jane discovers that the more complex Pareto-optimal candidates are refinements of a few simpler ones. She also perceives "Widows" as least frequent Marital Status (green).



3175	1250	872	34
670	6584	421	29
184	1128	1318	94
8	114	202	196

Accuracy [F1 score]:	0.687
Nr. of Leaves:	9
Nr. of Nodes:	17
Nr. of used Attributes:	4
Depth:	5
Avg. significant digits:	3.4
F1 Never married:	0.678
F1 Married:	0.785
F1 Divorced or Separated:	0.476
F1 Widowed:	0.449

Figure 4. Details for a selected tree evaluation. A node is hovered to focus on the explanation of *Widowed* persons.

5.4 Details for a selected model

Additional views of TreePOD show further details of the focal tree:

1) Structural aspects of the tree: when decision trees are used for explanatory purposes, inspecting the rules is essential. This includes the *names* of the used features, as well as their *depth* in the tree as a notion of their importance. Moreover, the exact split values are often important for explanation or hypothesis generation. The rule definition is also essential for qualitative judgments of interpretability based on domain knowledge. Another structural aspect of trees refers to the topology, e.g., distinguishing deep from wide trees.

To visualize these aspects, we use a node-link diagram inspired by BaobabView [46]. Each node contains its rule definition as text. The width of a link leading into a node is proportional to the number of data items that it applies to. Within links, space is subdivided into stacked, colored bands that convey the proportion of each class [46]. Since we want to emphasize the significance of paths and leaves, we reduce the visual footprint of other aspects. For example, we encode a leaf's decision as a colored triangle glyph instead of coloring the whole leaf, as this would result in large salient areas that distract from the significance of the links. For the same reason, we show detail information for nodes only on demand: when hovering a node, all nodes between it and the root show horizontally stacked bars conveying the gain of purity along the path (see Fig. 4). Hovering class labels in a coloring legend visually emphasizes leaves yielding that class.

As an indicator for decision confidence, we add a bubble to each leaf node, using the same pixel-based purity encoding as the plots in Sec. 5.3. Their size is proportional to the number of classified data items. Apart from making leaf nodes more salient, these bubbles facilitate visual correspondence of leaves with the tree map visualization.

2) Quantitative properties of the tree: The quantitative metrics listed in the beginning of Section 5 can be inspected in a list. In particular, this includes metrics currently not shown in the trade-off visualizations. As a familiar encoding of accuracy per class, we also provide a confusion matrix. A column-wise encoding of relative frequencies using a linear gray-scale informs the user about systematic misclassifications. On demand, users can switch to a row-wise relative encoding to focus on recall rather than precision. Absolute numbers are stated per cell to support comparisons in any case.

As TreePOD generates its tree candidates by parameter sampling, the particular parameter values that led to a tree can be interesting and are shown on demand. We hide this list by default to focus on the resulting trees, rather than the machine learning process (G2).

Inspecting the details of Pareto-optimal trees, Jane discovers "Age" as an important feature that is often used for the first split, mostly followed by "Sex", and "Income". "Age" seems to be important for the classification of Widow(er)s. The confusion matrix for the focal tree, however, reveals that less than half of all Widow(er)s are classified as such (bottom row in Fig.4b). She also discovers that the rule definitions are often not based on whole numbers, such as "Age > 27.5".

6 SENSITIVITY ANALYSIS OF TRADE-OFFS

Confidence in model selection is a multi-faceted topic. The visualizations described in the previous section provide no direct support for investigating how changes of the parameters involved in training, post-processing, and evaluation would affect the trees. This section describes extensions to the tree generation process and the visualization which enable an effective sensitivity analysis of parameter variations.

6.1 Generating tree families for effective comparison

Stochastic parameter sampling as described in Sec. 4 efficiently generates a diverse set of alternatives to choose from. However, these samples are usually too diverse to support a focused sensitivity analysis. As a solution, we extend the stochastic generation process by a controlled variation of one or more user-specified parameters, which are subsequently referred to as *variation parameters*. In contrast to other parameters, variation parameters are varied in a full-factorial manner and define a tree candidate for every possible combination of values. For each stochastic sample of the other parameters (Sec. 4), the controlled variation thus defines a *family* of trees. All members of one family are referred to as *sibling trees*. They only differ by the values of one or more variation parameters. For illustration, consider the variation of one parameter in the guiding example:

For her report, Jane prefers rules based on simple integer numbers, e.g. "Age > 28" rather than "Age > 27.5". She wonders if even multiples of 10 are sufficiently accurate. Thus, she varies the post-processing parameter "Round to significant digits" in three steps: {*"no rounding", "max. 2 significant digits", and "max. 1 significant digit"*}*. As a result, 3 variations are created for each of the 300 stochastic samples, which differ by the performed rounding. The new number of candidates is 900, comprising 300 families of 3 trees each.*

This two-step generation process ensures the existence of unbiased alternatives, and enables an effective assessment how a single tree, or the candidate set as a whole changes under controlled variations.

6.2 Sensitivity visualization

By default, the visualizations do not treat siblings differently from other possible candidates. As a result, one common Pareto-Front is computed, and shown in the quantitative and qualitative views.

TreePOD supports filtering the candidate set by variation parameters. In the candidate summary panel (Sec. 5.1), all values for each variation parameter are listed using labeled dot markers (see Fig. 1b). Clicking on a dot marker filters the set of visible tree candidates to those of the respective value. An additional marker labeled "any" does not filter on that parameter. Filters for multiple variation parameters are combined by a logical "AND". We refer to the vector of all current variation parameter values as the *variation focus*. Changing the variation focus updates the set of tree candidates which also updates the Pareto front. The corresponding sibling of the previous focal tree becomes the new focal tree, which also updates the detail visualizations.

For a sensitivity analysis regarding a specific variation parameter, the user may click on its name in the summary panel (e.g., "Round to significant digits" in Figure 1). The scatter plot then supports comparing the impact of parameter changes at three levels of locality.

1) Point-wise sensitivity of the focal tree. As the most local level, the scatter plot displays the siblings of the current focal tree as colored points. Inspired by previous work on sensitivity analysis [4], ordinal variations are connected by lines and encoded using different levels of luminance in the order of variation. For example, the turquoise points in Fig. 5f show how the focal tree changes for increasing maximal tree depths. For variation parameters without inherent order such as the pruning method, all siblings are connected to the focal tree. In this case, hue is used to discriminate the values. Our implementation attempts to use different hue sets for encoding data classes and variation values. This avoids color scheme overlaps if the numbers of classes and compared parameter values are low, which is a frequent case.

2) Point-wise sensitivity of Pareto-optimal trees. As a less local level, point-wise sensitivities can be shown for all currently visible Pareto-optimal trees. This enables to investigate how the sensitivity changes along the Pareto front. For example, Fig. 5d shows that evaluating trees for validation data leads to a stronger accuracy loss for complex trees than for simple ones.

3) Sensitivity analysis of the Pareto front. As the most global level of sensitivity visualization, the Pareto front itself is shown for each variation step. Each front is computed individually based on the candidates for the corresponding value of the investigated variation parameter. This enables a direct comparison of achievable trade-offs. In Fig. 1g, for example, the turquoise fronts indicate how the trade-off between accuracy and size changes for various rounding thresholds. The color scheme is the same as for point-wise sensitivity encoding.

6.3 Application to sub-tasks of model selection

The process of model selection comprises a number of sub-tasks which can be addressed by TreePOD. We identified four groups of tasks.

1) Sensitivity-aware selection of tree generation parameters

This group of tasks refers to studying the global effect of changing tree generation parameters. The focus of interest is typically on the achievable model characteristics and not on individual trees. Therefore, visualizing the entire front is typically the most suitable level of locality in this case. Typical goals include refining parameter ranges for the stochastic variation or assessing their stability for increased confidence. Specific examples for this group of tasks are:

Assessing the benefits of feature inclusion: Using features with high explanatory power is essential for a good fit, but some features may be expensive to obtain. Sometimes, these costs can be quantified, e.g., expensive medical tests [22]. Other times, they are subjective, such as side-effects of medical tests [45]. The latter are often only vaguely known and harder to compare across features. To support both types of costs, TreePOD enables a qualitative comparison of feature inclusion by varying whether a user-specified subset of the features is included. As an example, Fig. 5a shows the achievable Pareto fronts when including *Income-related* features in explaining Marital Status, or not. A reason to omit them could be a generally high number of missing values, when collecting such data from surveys.

Assessing accuracy loss due to decision border rounding: Rounding numerical decision thresholds in a post-processing step increases a tree's usefulness in human-oriented application contexts [15]. However, this typically decreases accuracy. Varying number rounding parameters, e.g., to *n* significant digits, supports the user in deciding how much accuracy should be sacrificed (see Fig. 5b).

Further examples refer to the variation of generation strategies, such as the feature selection criterion or the pruning method. For both parameters, several methods exist but no single one is considered generally superior [11, 30]. Visualizing the variation of Pareto fronts helps to understand the effect of different methods for the given dataset.

2) Assessment of model stability

From a statistical point of view, a weakness of decision trees refers to their high variance compared to other model types [14]. Slight changes in the training data may lead to substantially different model definitions. TreePOD supports an assessment of model stability by controlled variation of training data subsets. In this case, siblings refer to trees trained for the same parameters, but based on different data. When using meaningful data categories as subsets, encoding the Pareto fronts allows to identify categories for which classification is easier than for others. For example, the scatter plot in Fig. 5c shows that Marital Status is harder to predict for some ethnicities than for others.

3) Sensitivity of accuracy to changed evaluation data

Comparing model accuracy across different validation data subsets is a common approach for assessing generalizability to new data [14]. To enable such assessments, TreePOD supports a user-defined variation of the evaluation data subset analogous to the variation of generation parameters. In this case, siblings represent evaluations of the same tree for different data subsets. Showing these siblings for individual trees conveys how accuracy changes for different subsets, which supports the selection of robust models. Particular examples include:

Comparing training and validation data: Comparing tree evaluations for different training and validation data subsets provides guidance along the bias-variance trade-off. Fig. 5d, for example, shows a steadily increasing training accuracy, while the accuracy for validation data decreases for deeper trees due to over-fitting [14]. This provides effective guidance for selecting an adequate model complexity.

Comparing accuracy for data categories: Using meaningful data categories as evaluation subsets allows to identify a potential bias of the models, e.g. towards the most prevalent categories in the training

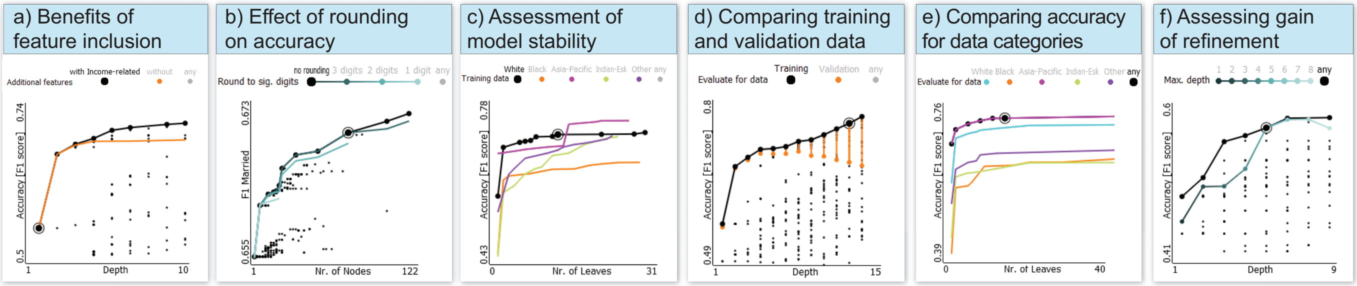


Figure 5. Various applications of a systematic variation of parameters in the generation, post-processing, and evaluation of decision trees. The sensitivity is shown for different levels of locality: the entire Pareto front (a, b, c, e), all Pareto optimal tree candidates (d), and a single tree (f).

data. Fig. 5e, for example, illustrates a variation of the evaluation data for different ethnic groups. The largest ethnic group of data records in the training data refers to "White" persons and also obtains more accurate classification than most others.

4) Building confidence in a selected tree TreePOD supports studying variations of a single tree to obtain confidence in its superiority. The point-wise sensitivity encoding is suitable for this purpose.

Assessing gain of refinement: By varying the termination criteria of tree construction (e.g. max. depth, or min. leaf size), TreePOD supports visualizing the benefits incurred by every split level. Reflecting the step-wise nature of the greedy construction process, the resulting line graph visualizes the construction history, to provide guidance for selecting an adequate depth. In Fig. 5f, for example, the variation of the maximal tree depth shows how the focal tree is not Pareto-optimal at first, but becomes part of the front after five refinement levels. Adding a split level increases the accuracy of the tree further, while 3 more levels yield a significant decrease for the validation data.

7 Workflow Integration

Building decision trees involves multiple steps [46]. The previous sections focused on the description of TreePOD for analyzing and choosing among an existing set of candidate trees. This section outlines the integration of TreePOD in a workflow for building decision trees. The subsequent steps are roughly ordered by their sequence in a typical workflow. However, our implementation does not enforce a particular order and permits most of them at any time.

Selecting training and validation data: Selecting plausible input data is typically a first step. In our implementation, users may interactively brush multivariate views of the data such as scatter plots, parallel coordinates, and time series plots to define data subsets for training and validation (see Sec. 8). Interactive data selection is useful, e.g., to exclude artifacts such as outliers based on domain knowledge. Alternatively, the system automatically defines disjunctive data sets for training and validation by random sampling of the input data.

Definition of initial tree candidates: On demand, TreePOD allows adjusting the variation strategy per generation parameter, i.e., fixed, stochastically sampled, or subject to a controlled variation (see Fig. 6). As the parameters have default values for sampling, users may also simply press a "Train" button to start without specifying parameters.

Global stochastic refinement of tree candidates: Initially, 300 stochastic variations are generated by default. Users may adjust this number depending on, e.g., the size of the training data and the number of features. At any time, users may then press a button titled "Show me more" to generate additional stochastic samples. For each of them, the same controlled variations are applied as for the initial set of trees. The set of Pareto-optimal trees will be re-evaluated for this new set, updating all views. This type of global augmentation of tree candidates is useful if the initial sampling turns out to be too sparse overall.

Local stochastic refinement of tree candidates: For a more focused, result-oriented refinement, users can create variants of the selected focal tree. Pressing a button titled "Show me more like this" will create new samples by stochastically varying the generation parameters such that they are similar to those of the focal tree, e.g., lying within narrow intervals for quantitative parameters. Repeating this for different Pareto optimal candidates allows steering the refinement of the front, and ensuring that interesting regions obtain enough samples. Alternatively, the user may inspect the particular parameter values for generating the focal tree. Users can then vary specific parameters while keeping all others fixed. For example, this enables to explicitly trigger the creation of additional hierarchy levels for a tree.

Extending the controlled variation: Users may specify or extend controlled variations of parameters at any time, e.g., if they identify interesting aspects for sensitivity analysis only after an initial inspection of the candidate trees. Each update of variation parameters is applied globally to all trees. This may generate new members of tree families or modify existing ones, e.g., if the controlled variation affects parameters which have previously been sampled stochastically.

Subjective validation of classification results: Clicking on any node of the focal tree as well as on rows and columns of the confusion matrix highlights the corresponding subset of training and validation data in the linked multivariate views. This supports a subjective validation of the classification results in the context of the actual data. In particular, this step may reveal if misclassifications are evenly distributed over the data or accumulate for, e.g., specific periods in case of time-dependent data or particular regions in case of spatial data. Sometimes, such findings may indicate structural breaks or insufficient quality for subsets of the data. Users can decide to exclude such subsets and re-run the generation for all models.

Extending the feature set: Detecting data subsets with many misclassifications may also inform domain experts about potentially missing features or may suggest the derivation of new features based on existing ones (e.g., decision boundaries defined by the interaction of multiple features). Derived features may, for example, be created in external computing environments and imported afterwards, e.g., from CSV files. Users may then either re-run the training for all tree candidates, or add the extended features as additional controlled variation.

Generation of sub-trees: It is sometimes helpful to focus the generation process on a particular sub-tree while considering other parts of the tree as given, e.g., if certain subsets of the data are more complex to model than others (Sec. 9 illustrates an example). In this case, users can specify a particular node of the focal tree as temporary root. This generates a set of candidates for this sub-tree using the same approaches for stochastic sampling and controlled variation as for the entire trees. Only these candidates are considered in this type of subtree mode. By default, only the data corresponding to the temporary root is considered for computation and visualization, and the result metrics refer to the sub-trees only. However, the structural tree still shows the position of the focal sub-tree within the entire tree as context (see Fig. 6). Upon leaving the sub-tree mode, the user may either add the focal sub-tree or all Pareto-optimal sub-trees as variants of the initiating focal tree to the overall set of tree candidates.

Local pruning of the focal tree: As the counterpart to growing sub-trees, users may also manually prune all nodes below a selected node of the focal tree. In contrast to automated pruning which is performed for all tree candidates, this type of local pruning is only applicable to the focal tree. The pruned tree is added to the set of candidates as a variation of the initiating focal tree.

8 Implementation

TreePOD has been implemented as a part of *Visplore*, a system for visual exploration of multivariate datasets. Visplore provides multiple linked views such as scatter plots, time series plots, and views for data categorization. Data subsets defined by brushing these views can be used in TreePOD as described in Sec. 7. The system is implemented in C++ and uses OpenGL for rendering. A multi-threading architecture [34] is used to maintain interactivity during computations.

For the identification of decision trees, we integrated the CART implementation of the open source library OpenCV [36]. Post-processing operations such as rounding are implemented on top of the tree definitions produced by OpenCV. In most cases, OpenCV was fast enough to generate large numbers of trees in a few seconds. Specifically, generating 300 trees for a data set of 32541 data records and 12 features took on average 5 seconds on a Desktop PC with Intel i7-2600k CPU at 3.4 Ghz and 16GB RAM. From a technical point of view, the ability to generate large numbers of trees rapidly is a key prerequisite for our approach and specifically the interactive workflow.

9 Evaluation

For evaluating TreePOD and the described workflow, we collaborated with four domain experts working for a transmission system operator and two experts from an IT service provider in the energy market. All of them have been active in this domain for multiple years. They are confronted with classification problems on a regular basis, e.g., for predicting market situations or for building treed prediction models of time series data. Nevertheless, all of them characterize themselves as having little background in statistical learning and very limited expertise with decision tree algorithms in particular. They used to address classification problems based on insights from static diagrams, intuition, and trial-and-error using common statistics software.

The evaluation took place in three workshops. In a first workshop, we introduced them for one hour to TreePOD by illustrating it based on three energy-related classification problems which were familiar to them from previous projects. They were allowed to ask questions at any time. Based on what they saw, the experts decided on a real-world classification problem as case study for a next workshop.

In this second workshop about one month later, we addressed that particular model selection problem (Sec. 9.1) after a brief recap of TreePOD. We strictly followed their instructions, but operated the software prototype ourselves. Two main reasons were limited time of the experts for familiarizing with all features, and the goal to keep them focused on aspects of the process rather than the implementation. Conducting the described case study took approximately one hour.

In a third workshop four months later, two of the experts used a deployed version of TreePOD to address a different model selection problem (Sec. 9.2.). This time, the experts controlled the system themselves, while we observed their actions and their workflow.

After each workshop, we asked the experts for their feedback using the rose-bud-thorn method [24] for another hour (Sec. 9.3).

9.1 Case study: prediction of imminent power shortages

The key task of power grid operators is to balance demand and supply of electricity. Volatile power sources such as wind farms, or fluctuations of energy prices may lead to spontaneous shortages or abundances in networks. Once such *critical situations* are in progress, they are expensive to fix. Recognizing their imminence in advance for early intervention can thus reduce financial costs significantly.

In a joint analysis session using TreePOD, domain experts identified decision trees predicting imminent critical situations. The target variable is a categorical time series with two classes "*critical in 15min*", and "*ok in 15min*", observed over 1 month ($\approx$ 260,000 records). Features comprise: (1) the DELTA between power supply and demand, (2) the used proportion of a limited RESERVE of balance energy, (3) various transformations of DELTA and RESERVE such as sliding averages over the past 10min (e.g. DELTA_10), first derivatives that express the TENDENCY of change, (4) 39 POWERPLANT production time series, and (6) categories such as MINUTE and HOUR.

For illustration, Fig. 6a shows examples of imminent critical situations, where RESERVE_10 is at its limit. The purpose of the model is to alert human decision makers rather than to replace them. In addition to high accuracy, having a small set of interpretable rules is thus considered highly important by the experts.

The experts initially select the first and second half of the observed time period as training and validation data. For generating an initial set of tree candidates, the experts stochastically vary the used termination criterion and the subset of input features to obtain 100 samples (see Fig. 6b). As variation parameter, the degree of rounding is varied in 4 steps. This results in 100 x 4 = 400 candidates.

The experts set accuracy and the number of nodes as objectives in the scatter plot. All tree maps of Pareto-optimal candidates show large, pure blue regions (Fig. 6c). Inspecting detail views reveals that critical situations are hardly ever imminent when $|\text{RESERVE\_10}|$ is below 76% of its limit (Fig. 6d). This is the first split of all Pareto optimal candidates. While this matches the expectation of the experts, the particular threshold value is relevant information for them. Classifying the remaining data, however, is more complex as shown by the noise at the margins of the increasingly complex tree maps. In order to focus the further analysis on explaining this remaining variance, the experts enter the sub-tree generation mode for the $|\text{RESERVE\_10}| \geq 76\%$ node. This creates a separate batch of 400 sub-tree candidates.

The visualizations of the Pareto-front now show 11 Pareto-optimal sub-tree candidates (Fig. 6e,f). In the scatter plot, the colored Pareto fronts for the varied degrees of rounding show that enforcing 3 or 2 significant digits does not incur a significant accuracy loss for smaller trees, while rounding to 1 digit does (Fig. 6e). After inspecting the trees in detail, the experts decide for 2 significant digits.

Browsing the Pareto-optimal candidates reveals that the feature TENDENCY_RESERVE is used for the first split by most sub-trees. This makes sense for the experts, as this feature indicates an increase (positive values) or decline (negative) of available balance energy.

By inspecting the Pareto front in the scatter plot, the experts soon decide for a sub-tree with two splits and an accuracy of approximately 0.73 (Fig. 6e). While the next simple candidate with a single split is much less accurate, significant gains of accuracy conversely require a much larger number of splits which contradicts the requirement for simplicity. The experts inspect further details for this selected focal tree (Fig. 6f,g). They are surprised that the second split by MINUTE has a threshold of 52, which they wish to investigate further. For this purpose, we configured an additional view of our system beyond TreePOD for the experts. Specifically, stacked bars show the proportion of critical situations per minute within the hour cycle. A click on the MINUTE-based split node in TreePOD updates the stacked bars to show only the corresponding data (Fig. 6h). This visualization confirms the adequacy of the split and also indicates a similarly blue region at the beginning of each hour. Based on this cyclic pattern, the experts hypothesize that the temporal proximity to the full hour might be an even more suitable feature than MINUTE. A composite brush for (MINUTE $>$52 OR MINUTE $<$ 5) enables to express HOUR CHANGE as a new binary input feature for TreePOD.

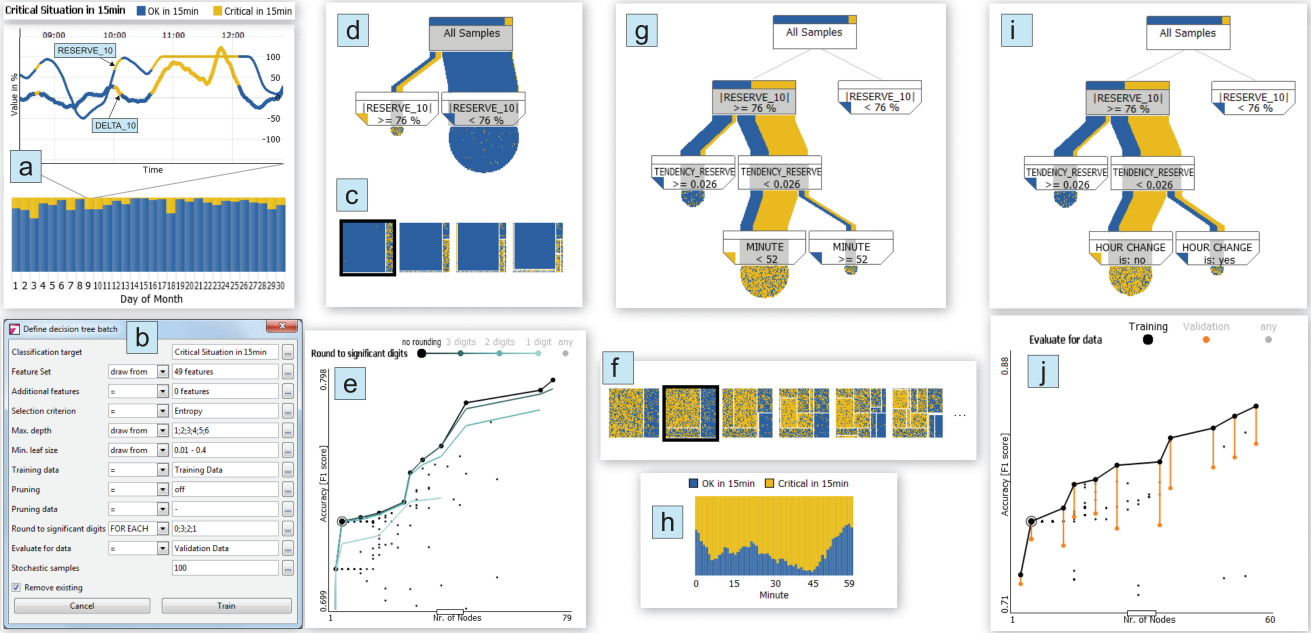
The experts specify an additional controlled variation regarding the inclusion of this feature. The point-wise sensitivity of the focal tree confirms an accuracy gain of approximately 2% for the corresponding sibling. This sibling also belongs to the updated set of Pareto optimal candidates and thus becomes the new focal tree (Fig.6i).

The experts are already very satisfied with this tree. As a final check, they want to validate its generalizability. A controlled variation of the evaluation data confirms the tree's accuracy for both training and validation data due to its relative simplicity (Fig.6j). More complex tree candidates are much less accurate for the validation data.

At the end of our joint session, the experts were very confident of having selected the most appropriate tree for their purpose. As a next step, they plan to test the performance in operation for a few weeks and eventually update the tree using TreePOD based on recent data.

9.2 Evaluation workshop with users of TreePOD

The third workshop took place four months later. After a brief recap, two of the experts controlled the system themselves for approximately one hour each, in individual sessions. The goal was to identify reasons for short-term changes of power production schedules, denoted as a categorical time series REDISPATCH (yes/no) over three months (2521 records). Features include 23 numerical time series represent-



Define decision tree batch

Classification target		Critical Situation in 15min
Feature Set	draw from	49 features
Additional features	=	0 features
Selection criterion	=	Entropy
Max. depth	draw from	1;2;3;4;5;6
Min. leaf size	draw from	0.01 - 0.4
Training data	=	Training Data
Pruning	=	off
Pruning data	=	-
Round to significant digits	FOR EACH	0;3;2;1
Evaluate for data	=	Validation Data
Stochastic samples		100

☑ Remove existing

Cancel Train

Figure 6. Predicting critical situations in power grid operation. Based on pre-classified data (a), varying decision tree generation parameters (b) obtains an ordered list of Pareto optimal model candidates (c). Inspecting the structure suggests an important first split (d). The Pareto fronts for different degrees of threshold roundness recommend a rounding degree of 2 (e) and a sub-tree candidate (f, g). The distribution of classification accuracy suggests adding the proximity to full hours as feature (h). The resulting tree (i) is better and generalizes for the validation data (j).

ing conditions of the network and the market, as well as temporal categories. This section describes how TreePOD was used by the experts. Screenshots of their insights can be found in the supplemental material. Feedback of the users is part of Sec. 9.3.

For an initial definition of tree candidates, the first user studied the dialog's options in depth first. She then started with sampling only the termination criteria, but provided all features to every tree as fixed assignment. The only difference between the resulting trees was their degree of refinement, allowing her to assess the benefits of splits. Surprised by the use of feature EXCHANGE_1 for the first split, she investigated alternative first splits by varying the features, while allowing just one split (max depth = 2). Browsing these trees revealed that no single split allowed splitting off a significant number of redispatch cases, and that the selection of EXCHANGE_1 as the first-split feature was justified. She then resorted to the default settings, and created a new batch of 100 trees based on sampling the features and termination criteria. Surprised by the high variation among the candidates, she repeatedly used the "Show me more like this" button to create more samples near the Pareto-front. She then spent some time browsing the fronts. A linked time series view highlighting periods classified as "REDISPATCH: yes" by the focal tree allowed her to compare the recognized redispatches across trees. Watching this view while browsing, she identified trees explaining the previously unexplained redispatch cases. This was a new way of exploration we had not tried before. She finally concluded that the redispatch periods during the first two months can be classified well using trees of moderate depth (≤ 4), which she considered useful for reporting. Trees that also explain the periods of the last month, however, require significantly more nodes.

The second user defined an initial batch of 500 candidates by sampling the features and termination criteria. Browsing the Pareto-optimal trees enabled him a quick identification of important features, as well as a preferable tree depth (max 4-5) for reporting. Like the first user, he was curious about alternative explanations without the dominant feature EXCHANGE_1. Thus, he extended the candidates by controlled variation of omitting vs. providing this feature to the trees. He discovered that a related feature NET_1 is often selected as a substitute, resulting in trees with comparable accuracy. He then used the same linked time series view as the first user while browsing the trees. He hypothesized that the cause for redispatch periods might have changed after the second month. Thus, he decided to split the data sets based on this possible structural break, and trained trees for each part individually. He discovered that trees for the third month did not use EXCHANGE_1, but rather four other features, confirming his hypothesis. Finally, controlled variation of border rounding showed him that rounding to 3 or even 2 significant digits incurs little accuracy loss for most trees, which he appreciated for his report.

In conclusion, both experts were satisfied with the explanations they found, and considered them useful for their reports.

9.3 Qualitative Feedback

The six domain experts stressed the importance of building classification models as part of their jobs. Some models need to be updated frequently due to rapid changes in the energy sector. Consequently, the time they can spend on tuning single models is limited (G3). Moreover, they believe that many domain experts in their field lack a deep statistical knowledge (G2). For all six experts, model accuracy and complexity are typically the most important aspects. Other requirements such as feature acquisition costs and model plausibility need to be considered as well, but are often hard to quantify. Thus, they appreciated that the controlled variation allowed them to compare discrete sets of model variants without the need for quantification.

The reaction of the experts to TreePOD was very positive overall. All of them praised the possibility of getting a fast overview of possible model characteristics as a huge step forward in comparison to their current practice (G1). In particular, all experts considered the knowledge about the variability of model characteristics and achievability of model objectives as significant gain of confidence (G4). The result-driven approach was embraced as very understandable. The detail visualizations of the model were considered crucial both for understanding the approach as such and for supporting a qualitative model assessment. In general, all experts claimed to have understood TreePOD within the first workshop to a degree which enabled them to think about applications to own classification problems. We specifically asked them if they consider the controlled variation as beneficial without deep algorithmic knowledge. Four experts answered that important variation options do not require such knowledge in their opinion, e.g., the set of input features or rounding levels. Two of the domain experts also considered the variation of other generation parameters as helpful for non-experts in statistics to develop an intuition of their impact.

When asked about specific visualizations, five experts considered the tree maps as important intermediate level of complexity between the abstract scatter plot and the detailed structural visualization. They considered their linear order as an intuitive guidance through the candidates. However, all experts agreed that the scatter plot is crucial as an overview and for conveying the shape of the Pareto front, e.g., for an efficient perception of jumps and sparsely sampled regions.

As a shortcoming, two experts questioned the restriction to binary trees, i.e., each intermediate node having two children. Despite advantages of binary trees from a statistical point of view [14], they considered more general trees as easier to understand and to communicate, e.g., when subsequent splits refer to the same feature.

The experts who used TreePOD themselves found the default sampling parameters combined with the “Show me more like this” button highly enabling for users without statistical background. However, they considered the number of 20 added samples with every press of this button inadequately small. One expert considered a time-based specification a better alternative, e.g., sampling for 1-2 seconds. One expert suggested adding dedicated buttons to trigger important variations more easily, e.g., “create rounded variations”, or “omit feature”. When defining filters on result metrics, one expert suggested drawing the achievable Pareto front for the filtered trees as context. Concerning the bubble encoding of leaf nodes (see Fig. 4a), the users found purity better conveyed by the stacked bars and bands between nodes. However, one user said their correspondence to the tree maps helped to understand the latter visualization, which was unfamiliar at first.

The other experts also contributed numerous ideas for further extensions. One expert stressed that upper hierarchy levels should be definable from the outside in order to represent given (political) rules and classification schemes. Another expert requested a sensitivity analysis for decision thresholds of particular nodes. As a very interesting idea, one expert suggested using TreePOD to explain user-defined data subsets. For example, after brushing an anomalous period of energy production in a time series view, TreePOD could explain this period by other time series such as meteorological conditions.

10 Discussion and Future Work

TreePOD fosters a shift in the strategy for tuning the generation parameters of decision trees. Fully automated tree generation often results in a cumbersome trial-and-error parameter search [32]. Most previous work for cooperative decision tree construction [3, 23, 46] follow a local-to-global strategy for investigating the parameters [40]. These approaches can be classified as white-box integration of visualization and mining [5]. In contrast, TreePOD can be considered a black-box type of integration. A key advantage is to hide details of the generation process from users unless on explicit request (G2). Moreover, TreePOD encourages a global-to-local search strategy which starts with an overview of possible characteristics for reducing the risk of missing the global optimum (G1). TreePOD still supports a cooperative creation, but on a global scale rather than by focusing on a single tree. Specifically, controlled variations are applied to the entire set of candidates which enables a comparison of the effect across trees for higher user confidence (G4). However, this concept does not exclude local refinements of selected trees if explicitly requested by users (see Sec. 7).

TreePOD closely follows the Visual Analytics Mantra [17]: To *analyze first*, TreePOD generates a comprehensive set of decision candidates and computes quality metrics for them. TreePOD *shows the important* by focusing the selection on Pareto-optimal tree candidates. Users may *zoom and filter* by quality metrics. Adding tree candidates enables to *analyze further* for inspecting sensitivities regarding controlled variations of the tree generation parameters as well as for refining the sampling towards desirable tree properties. Additional views provide *details on demand* for a selected tree.

An important design decision of TreePOD is to restrict the number of Pareto objectives to two. This limitation has several significant advantages for keeping the approach understandable by users. For visualization, the simple representation as poly-line permits an intuitive comparison of variations of the entire front. For interaction, the linear order of tree candidates along the trade-off enables an intuitive switch from one tree to the next more accurate or more simple Pareto-optimal tree. For guidance in general, the set of Pareto optimal tree candidates is typically much smaller for two objectives than for three or more objectives, which avoids overwhelming the user with too many alternatives (G3). Moreover, feedback by domain experts suggests that the trade-off between accuracy and complexity is typically the most important consideration, even if additional objectives such as feature acquisition cost exist. Additional objectives can be considered by filtering trees with undesirable values as a common approach to address multi-criteria decision problems [44]. Nevertheless, experimenting with visualization approaches for higher-dimensional multi-criteria decision making [33] is relevant as future work.

Regarding other scalability aspects, the use of hue restricts the number of target classes to approximately ten for perceptual reasons [47]. Even more so, as color is also used for encoding the variation. We also experimented with showing variations of multiple parameters simultaneously, but rejected this feature due to generating too complex visualizations in many cases. On the other hand, the visual complexity of TreePOD does not depend on the size and dimensionality of the training or validation data. As a practical limit of the data size, however, the approach strongly benefits from short training times of trees in order to generate a sufficiently dense sampling overall and of the Pareto front in particular. The quantitative overview scales well for large numbers of trees, considering that the most relevant information is the location and shape of the Pareto front. Conversely, a sparse sampling will in general obtain a very inaccurate approximation of the real Pareto front. While local refinements of the sampling help to mitigate this problem (Sec. 7), integrating advanced approaches for constructing the Pareto front [9, 50] are an important aspect of future work.

As a next step, we plan to conduct a long-term study based on deploying TreePOD to target users from multiple application domains. Moreover, we intend to extend the approach in order to further utilize information contained in the generated set of candidate trees. For example, analyzing the frequency and the context in which particular features are selected could provide useful information about their importance. Finally, we believe that core concepts of TreePOD are transferable to other types of models. Model selection is typically a multi-criteria problem. In addition to accuracy, objectives regarding, e.g., comprehensibility and feature acquisition cost apply to many types of models [12], e.g., regression polynomials. We thus plan to evaluate in how far the concepts of TreePOD regarding sampling, guidance, and variation also support the selection process for other types of models by replacing decision tree-specific result metrics and visualizations.

11 Conclusion

This paper described TreePOD, a new approach for sensitivity-aware selection of decision trees in the presence of multiple objectives. Besides accuracy, especially the need for comprehensible models is increasing [19]. To address this need, TreePOD fosters a global-to-local strategy for model selection in order to guide also non-experts in statistical modeling towards a confident selection of suitable trees.

Based on TreePOD, we described a holistic workflow for decision tree selection which combines aspects from white-box and black-box integration of visualization and data mining [5]. A case study conducted in pair-analysis with domain experts illustrated the ability of TreePOD to solve a relevant problem in the energy sector, and confirmed that non-experts in statistics were able to efficiently identify a suitable decision tree with high confidence. TreePOD is applicable to classification problems independent of the application domain. As one possible direction of future work, we believe that TreePOD is conceptually transferable to other types of models for increasing the efficiency and confidence in the selection process.

Acknowledgments

This research is funded by the COMET K1 program – Competence Centers for Excellent Technologies (854174) by BMVIT, BMWFW, Styria, Styrian Business Promotion Agency – SFG and Vienna Business Agency. This work has also been supported by the Austrian Funding Agency (FFG) within the scope of the K-project DEXHELPP (843550). The COMET Programme is managed by FFG. Thanks go to all collaborators from the energy sector, and to C. Arbesser, O. Rafelsberger, S. Pajer, and Torsten Möller’s group for valuable comments.

References

[1] S. Afzal, R. Maciejewski, and D. Ebert. Visual analytics decision support environment for epidemic modeling and response evaluation. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pages 191–200, 2011.

[2] 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 Conf. on Knowledge Discovery and Data Mining*, KDD '99, pages 392–396, NY, USA, 1999. ACM.

[3] M. Ankerst, M. Ester, and H.-P. Kriegel. Towards an effective cooperation of the user and the computer for classification. In *Proceedings of the Sixth ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, KDD '00, pages 179–188, NY, USA, 2000. ACM.

[4] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. In *Computer Graphics Forum*, volume 30, pages 911–920, 2011.

[5] E. Bertini and D. Lalanne. Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. *ACM SIGKDD Explorations Newsletter*, 11(2):9–18, 2010.

[6] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. CRC Press, New York, 1999.

[7] S. Bruckner and T. Möller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1467–1475, 2010.

[8] W. Castillo-Rojas, C. Vargas, and C. M. Villegas. Interactive visualization of association rules model using SOM. In *Proceedings of the XV International Conference on Human Computer Interaction*, page 104, 2014.

[9] M. Czajkowski and M. Kretowski. A multi-objective evolutionary approach to pareto optimal model trees. a preliminary study. In *Theory and Practice of Natural Computing*, pages 85–96, 2016.

[10] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 155–164, 1999.

[11] F. Esposito, D. Malerba, G. Semeraro, and J. Kay. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):476–491, May 1997.

[12] M. Gleicher. A framework for considering comprehensibility in modeling. *Big Data*, 4(2):75–88, 2016.

[13] J. Han and N. Cercone. Interactive construction of decision trees. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 575–580, 2001.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Second Edition*. Springer New York Inc., 2009.

[15] J. Jung, C. Concannon, R. Shroff, S. Goel, and D. G. Goldstein. Simple rules for complex decisions. *CoRR*, arXiv:1702.04690, 2017.

[16] G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied statistics*, pages 119–127, 1980.

[17] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. In S. Simoff, M. Böhlen, and A. Mazeika, editors, *Visual Data Mining*, pages 76–90. Springer, 2008.

[18] M. M. Köksalan, J. Wallenius, and S. Zionts. *Multiple criteria decision making from early history to the 21st century*. World Scientific, 2011.

[19] H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684, 2016.

[20] P. Lévy. *Pixelization Paradigm: Visual Information Expert Workshop, VIEW 2006, Paris, April 2006, Revised Selected Papers*. Image Processing, Computer Vision, Pattern Recognition, and Graphics. Springer, 2007.

[21] M. Lichman. UCI machine learning repository, 2013.

[22] C. X. Ling, V. S. Sheng, and Q. Yang. Test strategies for cost-sensitive decision trees. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1055–1067, 2006.

[23] Y. Liu and G. Salvendy. Design and evaluation of visualization support to facilitate decision trees classification. *International Journal of Human-Computer Studies*, 65(2):95–110, 2007.

[24] LUMA Institute. Vision Statement: A Taxonomy of Innovation, 2014.

[25] K. Matković, D. Gračanin, M. Jelović, and H. Hauser. Interactive visual steering—-rapid visual prototyping of a common rail injection system. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1699–1706, 2008.

[26] C. J. Meneses and G. G. Grinstein. Visualization for enhancing the data mining process. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 126–137, 2001.

[27] T. Mühlbacher, H. Piringer, S. Gratzl, M. Sedlmair, and M. Streit. Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1643–1652, Dec 2014.

[28] E. Müller, I. Assent, R. Krieger, T. Jansen, and T. Seidl. Morpheus: interactive exploration of subspace clustering. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1089–1092, 2008.

[29] T. Munzner. *Visualization analysis and design*. CRC Press, 2014.

[30] S. K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data mining and knowledge discovery*, 2(4):345–389, 1998.

[31] K.-M. Osei-Bryson. Evaluation of decision trees: a multi-criteria approach. *Computers and Operations Research*, 31(11):1933 – 1945, 2004.

[32] L. Padua, H. Schulze, K. Matković, and C. Delrieux. Interactive exploration of parameter space in data mining: Comprehending the predictive quality of large decision tree collections. *Computers & Graphics*, 41:99 – 113, 2014.

[33] S. Pajer, M. Streit, T. Torsney-Weir, F. Spechtenhauser, T. Möller, and H. Piringer. Weightlifter: Visual weight space exploration for multi-criteria decision making. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):611–620, 2017.

[34] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A multi-threading architecture to support interactive visual exploration. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1113–1120, Nov. 2009.

[35] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. R. Johnson. Ensemble-vis: A framework for the statistical visualization of ensemble data. In *IEEE Conference on Data Mining Workshops*, pages 233–240, 2009.

[36] K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov. Real-time computer vision with opencv. *Comm. of the ACM*, 55(6):61–69, 2012.

[37] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[38] J. R. Quinlan. Improved use of continuous attributes in c4.5. *Journal of artificial intelligence research*, 4:77–90, 1996.

[39] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proc. of the Fifth Int. Conf. on Coordinated and Multiple Views in Exploratory Visualization*, pages 61–71, 2007.

[40] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014.

[41] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on graphics (TOG)*, 11(1):92–99, 1992.

[42] J. Talbot, S. Lin, and P. Hanrahan. An extension of wilkinson's algorithm for positioning tick labels on axes. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2010.

[43] T. Torsney-Weir, A. Saad, T. Möller, H.-C. Hege, B. Weber, and J.-M. Verbavatz. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):1892–1901, 2011.

[44] T. Torsney-Weir, M. Sedlmair, and T. Möller. Visualization for decision making under uncertainty. In *Workshop on Visualization for Decision Making under Uncertainty (VDMU)*, 2015.

[45] P. D. Turney. Types of cost in inductive concept learning. *CoRR*, cs.LG/0212034, 2002.

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

[47] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., 2004.

[48] M. Ware, E. Frank, G. Holmes, M. Hall, and I. H. Witten. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292, 2001.

[49] H. Wickham, D. Cook, and H. Hofmann. Visualizing statistical models: Removing the blindfold. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8(4):203–225, 2015.

[50] H. Zhao. A multi-objective genetic programming approach to developing pareto optimal decision trees. *Decision Support Systems*, 43(3), 2007.

[51] X. Zhu and I. Davidson. *Knowledge discovery and data mining: challenges and realities*. Premier reference source. Information Science Reference, 2007.