Computers & Geosciences 170 (2023) 105248

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

# Integrating scientific knowledge into machine learning using interactive decision trees

Georgios Sarailidis [a,*], Thorsten Wagener [b], Francesca Pianosi [a]

[a] Water and Environmental Engineering, Department of Civil Engineering, University of Bristol, Bristol, United Kingdom
[b] Institute of Environmental Science and Geography, University of Potsdam, Potsdam, Germany

ARTICLE INFO

*Keywords:*
Interactive decision trees
Machine learning
Human-in-the-Loop
Interpretability
Open-source toolbox
Geosciences and environmental sciences

ABSTRACT

Decision Trees (DT) describe a type of machine learning method that has been widely used in the geosciences to automatically extract patterns from complex and high dimensional data. However, like any data-based method, the application of DT is hindered by data limitations, such as significant biases, leading to potentially physically unrealistic results. We develop interactive DT (iDT) that put humans in the loop to integrate the power of experts' scientific knowledge with the power of the algorithms to automatically learn patterns from large datasets. We created an open-source Python toolbox that implements the iDT framework. Users can interactively create new composite variables, change the variable and threshold to split, prune and group variables based on their physical meaning. We demonstrate with three case studies how iDT overcomes problems with current DT thus achieving higher interpretability and robustness of the result.

## 1. Introduction

In the past few decades, our ability to collect, store and access large volumes of earth systems data has increased at unprecedent rates thanks to improved monitoring and sensing techniques (Hart and Martinez, 2006; Butler, 2007; Karpatne et al., 2017; Zhou et al., 2017), ever growing computational power (Washington et al., 2009), and the development of simulation models that produce large datasets at increasing domain scale and resolution. An example is the CMIP-5 dataset of the Climate Model Intercomparison Project (which contains various climatological variables at daily resolution (1980–2300) with global coverage and over 3 petabytes in size) that has been used extensively for scientific groundwork towards climate assessments (Reichstein et al., 2019). This 'data deluge' has paved the way for the systematic processing and analysis of observational and simulation data, often using Machine Learning or other statistical methods (Reichstein et al., 2019; Karpatne et al., 2019; Sun et al., 2022).

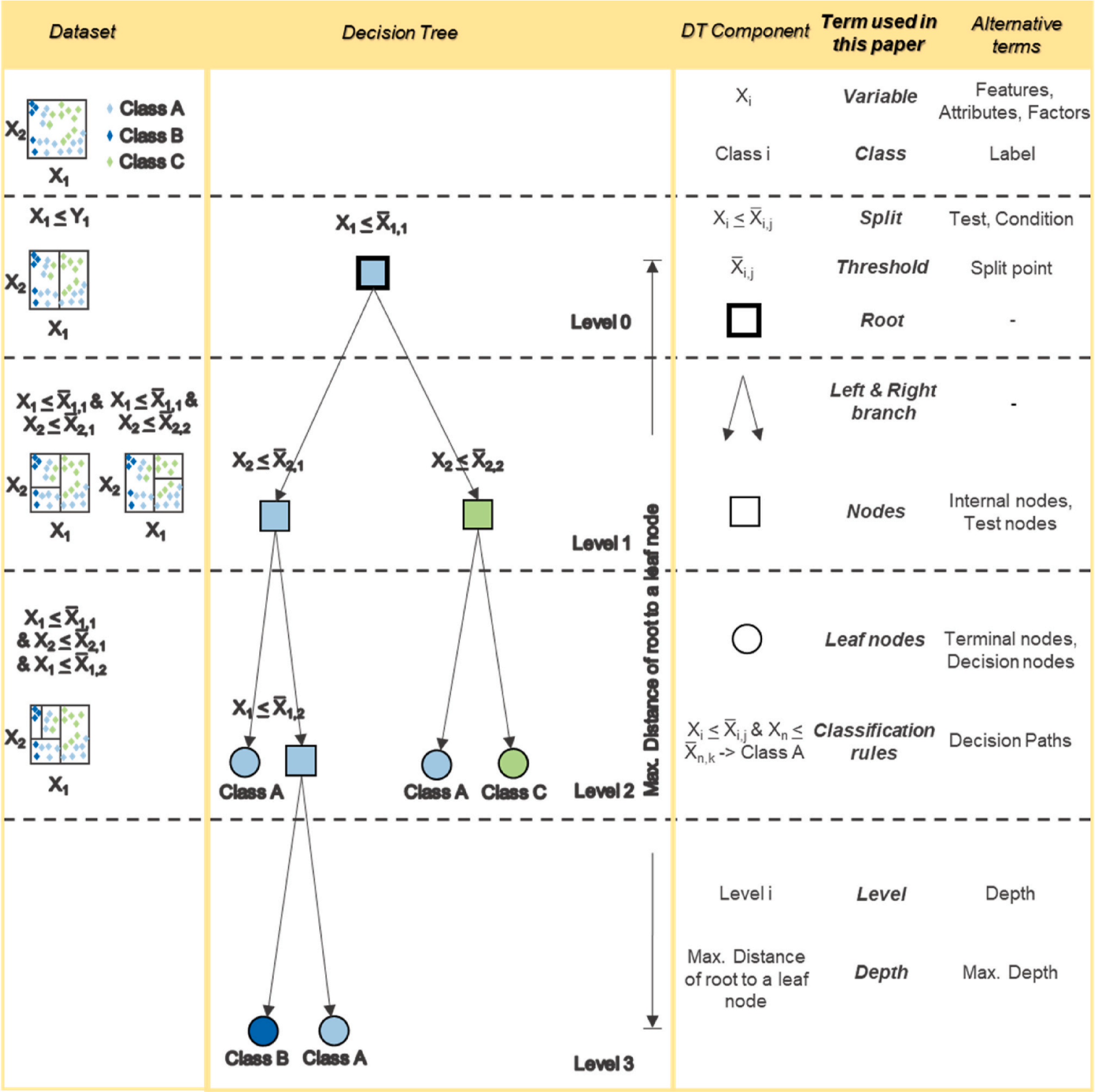
Machine Learning (ML), a term defined by Samuel (1959), is a branch of artificial intelligence (AI) and computer science which focuses on discovering patterns hidden in complex datasets (Bzdok et al., 2017; Reichstein et al., 2019) by imitating the way that humans learn (IBM, 2020). The main purpose of ML is to develop algorithms that can learn from historical data and perform tasks (e.g. predictions and classification) on new input data. The capability of ML methods to automatically extract patterns from large volumes of complex and high-dimensional data have made them an important part of research in many fields, including the geosciences (Bergen et al., 2019; Sun et al., 2022).

In this paper we focus on a method called Decision Tree (DT) (Breiman et al., 1984), a supervised ML method that is widely used in the geosciences. A DT model is developed through an automatic algorithm that recursively partitions the space of input variables into subspaces using a set of hierarchical decisions. In Fig. 1, we show a DT with a schematic representation of the recursive partitioning of the dataset along with basic terms used in this paper. A DT model is a hierarchical tree structure that comprises nodes and branches. Each node is associated with a logical expression, i.e. a "split", which consists of the variable and threshold to split, e.g. "$X_i$ smaller than $\overline{X}_{i,j}$". Each node will lead to two branches that correspond to the different possible outcomes of the split. The terminal nodes are called leaves and are associated to either a class or a specific value for the output. DT are thus commonly used for (Flach, 2012):

- Classification: The DT is trained on output data that are categorized under different classes (discrete values or non-numerical categories) and can predict classes for unseen data.

[*] Corresponding author.
*E-mail address:* g.sarailidis@bristol.ac.uk (G. Sarailidis).

https://doi.org/10.1016/j.cageo.2022.105248
Received 29 March 2022; Received in revised form 7 August 2022; Accepted 9 October 2022
Available online 15 October 2022



| DT Component | Term used in this paper | Alternative terms |
|---|---|---|
| $X_i$ | Variable | Features, Attributes, Factors |
| Class i | Class | Label |
| $X_i \le \bar{X}_{i,j}$ | Split | Test, Condition |
| $\bar{X}_{i,j}$ | Threshold | Split point |
| □ | Root | - |
| ↙↘ | Left & Right branch | - |
| □ | Nodes | Internal nodes, Test nodes |
| ○ | Leaf nodes | Terminal nodes, Decision nodes |
| $X_i \le \bar{X}_{i,j}$ & $X_n \le \bar{X}_{n,k}$ -> Class A | Classification rules | Decision Paths |
| Level i | Level | Depth |
| Max. Distance of root to a leaf node | Depth | Max. Depth |

**Fig. 1.** Left: A schematic representation of the recursive partitioning of the data space performed by a Decision Tree development algorithm. Middle: A typical Decision Tree. Right: Terminology.

- Regression: The DT is trained on continuous output variables, and it predicts continuous values instead of classes.

Examples of DT applications in the geosciences include, catchment classification (Sawicz et al., 2014; Kuentz et al., 2017), land cover classification (Gislason et al., 2006), studying uncertain factors of simulation models (Almeida et al., 2017; Sarrazin, 2018), analyzing rainfall-runoff relationships (Iorgulescu and Beven, 2004; Singh et al., 2014), empirical streamflow simulation (Shortridge et al., 2016), soil mapping (Grimm et al., 2008; Hengl et al., 2017), regionalizing hydrological signatures (Addor et al., 2018).

DTs are quite appealing in the geosciences because geophysical processes often reveal hierarchical structures of controlling variables, and the hierarchical structure of DT with nodes, branches and splits is a straightforward way to capture this. In geoscience applications, DT are particularly appealing for the purpose of organizing spatially distributed entities, such as rivers, catchments or other landscape units, thus demonstrating how large-scale (e.g. climatic) controls interact with small-scale (e.g. land use or geology) controls (Sawicz et al., 2014; Addor et al., 2018).

Despite these advantages, they have limitations (see Fig. 2) which make their use in the geosciences challenging. We highlight three main challenges that are important to our discussion:

1) Like any statistical tool, DT methods rely on data and consequently their credibility is dependent on the quantity and quality of data available. DT require large amounts of data for training which are not always available (Kirchner et al., 2020). When available, data in geosciences can be biased, complex, uncertain, noisy, heterogeneous and with changing properties (e.g. due to changes in measurement

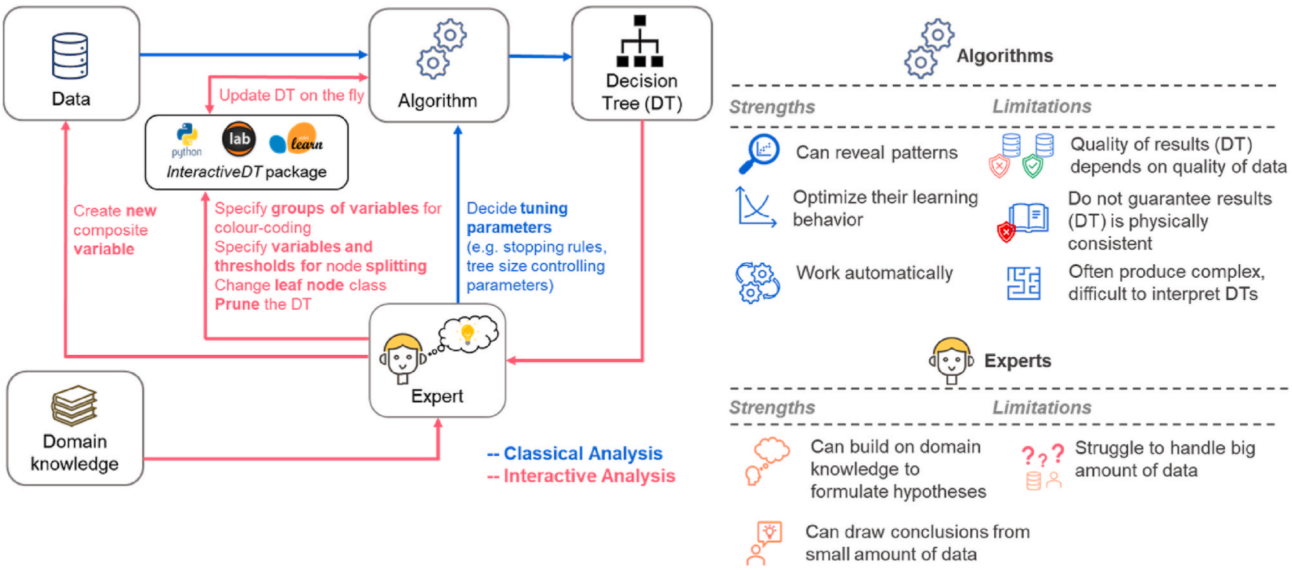

Data
Update DT on the fly
Algorithm
Decision Tree (DT)
python
lab
learn
InteractiveDT package
Create new composite variable
Specify groups of variables for colour-coding
Specify variables and thresholds for node splitting
Change leaf node class
Prune the DT
Decide tuning parameters (e.g. stopping rules, tree size controlling parameters)
Expert
Domain knowledge
-- Classical Analysis
-- Interactive Analysis
Algorithms
Strengths
Limitations
Can reveal patterns
Quality of results (DT) depends on quality of data
Optimize their learning behavior
Do not guarantee results (DT) is physically consistent
Work automatically
Often produce complex, difficult to interpret DTs
Experts
Strengths
Limitations
Can build on domain knowledge to formulate hypotheses
Struggle to handle big amount of data
Can draw conclusions from small amount of data

**Fig. 2.** **Left:** Flowchart of the steps performed to develop a Decision Tree in a "Classical" analysis and with our proposed Interactive analysis. **Right:** Strengths and Limitations of Decision Trees Algorithms and experts.

instruments or the data processing algorithms) (Solomatine and Ostfeld, 2008; Faghmous and Kumar, 2014; Beven et al., 2018; Karpatne et al., 2019). Therefore, the accuracy of DTs deteriorates with decreasing size or quality of the training dataset (Pal and Mather, 2003).

2) DT development relies on statistical metrics and algorithmic decisions aim at statistical optimality, usually measured in terms of classification rate or regression accuracy. However, such statistical optimality does not guarantee that the outcome is physically consistent (Roscher et al., 2020). By physical consistency we mean that a DT should not violate scientific principles (such as conservation of mass), or overlook known physical characteristics of the system investigated. For example, some input variables may have physically meaningful threshold values that may be missed by the DT because other threshold values might produce a statistically better result for the (noisy and biased) dataset used for training. Moreover, most DT algorithms use split rules based on a single variable at each node, whereas combinations of multiple variables may play a significant role in partitioning the data space (Loh, 2014; Almeida et al., 2017).
3) DT complexity may decrease their interpretability and consequently limit their usefulness in geosciences applications. By interpretability we mean the ability by a human expert of making sense of the obtained model (Molnar, 2020), understand how the model works and reaches a specific decision. Decision Trees are easier to interpret if they are small. The greater the number of terminal nodes, the deeper the tree and the more difficult it becomes to interpret. (Molnar, 2020; Lipton, 2018). Visualization could also help increase the interpretability of DTs. However, existing visualization techniques mainly focus on displaying information related to the statistical properties of the DT (e.g. impurity, node data points), whereas they do not support the display of information related to the physical properties of the variables – something that would potentially be more useful for geosciences applications (Almeida et al., 2017).

Integration of human experts in the DT development process – and hence of their domain knowledge and their cognitive ability to formulate hypotheses and theories – may help overcome some of these challenges. For example, experts may have very good knowledge of the physical processes, quantities and phenomena under study and hence be able to define physically meaningful splitting variables and thresholds, or discard DT branches that are physically unrealistic. An example is given in Stein et al. (2020) where a DT model to classify river flood generating processes is built purely based on domain knowledge. In addition, experts can define combinations of input variables that they believe interact in controlling outputs, where current algorithms would not allow for the detection of such combinations. An example is given in Almeida et al. (2017), where expert knowledge enabled integrating multiple variables into a new and physically meaningful factor. Moreover, experts can learn patterns from few datapoints because they have a certain expectation of relevant causal relationships, so they could guide the algorithm to learn from smaller amounts of data, or dataset where a particular output class is under-represented ("imbalanced dataset" (García and Herrera, 2009). Inclusion of domain knowledge in the model building process can also increase trust in the modelling results (Solomatine and Ostfeld, 2008). Incorporating scientific knowledge into ML models to improve their physical realism and interpretability has been highlighted as a major challenge and opportunity for ML applications in the geosciences (Read et al., 2019; Sun et al., 2022).

In this paper, we propose a framework to develop "interactive Decision Trees" (iDTs) that put human experts in the development loop of Decision Trees. Our iDT framework establish a two-way interaction between the automatic DT development algorithm and the expert, allowing the expert to manually create new composite variables, changing nodes' splitting variables and thresholds, manually pruning leaf nodes, and visualizing DTs in physically meaningful ways. Past attempts at developing iDT include the works of Ankerst et al. (2000), Han and Cercone (2001), Teoh and Ma (2003), Fails and Olsen (2003), Solomatine and Siek (2004), Mickens et al. (2007), Do (2006), van den Elzen and van Wijk (2011), Estivill-Castro et al. (2020), Elia et al. (2021). Outside the scientific literature we found two commercial software products that allow users to interact with the DT, Dataiku[1] and IBM SPSS.[2] We discuss briefly why the additional work presented here is warranted despite these previous efforts:

1. To our understanding, the above tools were developed for general use and none of them was tested for geosciences applications. This puts into question whether their interactive functionalities will be

[1] https://www.dataiku.com/.
[2] https://www.ibm.com/analytics/spss-statistics-software.

applicable and/or useful to overcome the specific challenges discussed above. For example, we ran the tool developed by van den Elzen and van Wijk (2011), but it could not handle the large datasets typical for geoscience applications. The tool by Solomatine and Siek (2004), which was tested on six hydrological datasets, allows for larger datasets, but it is not publicly available.

2. To the best of our knowledge, none of the studies cited above publicly shared the code to run their analyses and this might be one reason why they were not followed up by others or adopted by researchers in our community. The exceptions are: (a) the web application from Elia et al. (2021), which is freely available open source, but is designed for educational purposes; (b) the commercial software Dataiku, which is freely available for academic purpose but not open-source; and (c) the IBM software, which is neither free nor open-source.
3. Finally, in the above tools the main purpose for integrating human expertise in the DT development process is to improve the algorithm' predictive performance. Here instead we argue that interpretability and robustness are at least equally important aspects in geosciences applications. Even to the point that users might accept a reduction in predictive performance if it comes with an increase in interpretability and robustness given new datasets. Hence, we devise and demonstrate a number of visualization and interaction functionalities that are specifically aimed at increasing DT interpretability, and we also discuss how to measure interpretability (in the context of a specific application – see Case Study 2) beyond simply measuring the DT size (number of layers, number of leaf nodes) and training time (as done in previous studies).

Alongside presenting our iDT framework, we thus also introduce a free, open-source Python package to implement the iDT framework, which we demonstrate using three case studies representatives of typical challenges encountered in the geosciences. In the first case study, we show how color-coding the tree nodes based on their physical meaning produce a physically meaningful visualization, and how the experts can create new composite variables in the training process to better capture existing interactions in the dataset, thus producing a smaller and more interpretable DT. In the second case study, we show how the expert can manually change the splitting threshold values of the tree nodes based on other sources of knowledge, again to increase the DT interpretability. Finally, in the third case study, we show how experts can manually change nodes' splitting variables and thresholds to include under-represented classes in imbalanced datasets and make the DT physically consistent, robust and potentially even more accurate on new datasets.

## 2. Methodology

In this section we describe our framework for establishing interactions between the expert and an automatic DT training algorithm to integrate scientific knowledge in DT development. Moreover, we describe the Python package and the Jupyter Lab Graphical User Interface we developed to implement the framework. Finally, we present our ideas on how to evaluate DT predictive and interpretive performance.

### *2.1. A framework for interactive construction and analysis of decision trees*

Fig. 2 shows our framework for interactive construction and analysis of DTs and compares it to the classical approach of automatic development. In the classical approach, the analyst prepares the dataset to feed to the ML algorithm, specifies the algorithm's tuning parameters, executes it, and obtains the classification/regression model. In the interactive framework, the analyst (expert) can input their prior knowledge and/or feedback. Specifically, the expert can:

1) Organize and (pre-)process the input datasets, by grouping input variables in a physically meaningful way (such as climate variables, land surface properties, soil properties, etc.). The tree can then be colour coded based on this grouping. The user can also add new composite variables to the input dataset before or after the first algorithm run.
2) Directly manipulate the structure of the DT model, by changing node variables and split threshold values, or by manually pruning the DT or changing leaf node class. This can be useful when the expert is aware of physically meaningful threshold values for certain variables (for example thresholds for climate variables that are used to classify different climate zones) to improve the DT's physical interpretability. Another reason to manipulate the DT structure is the case of an imbalanced dataset, where a certain class is under-represented in the dataset and thus an automatic algorithm may not separately represent that class in the DT. Different tactics have been proposed to overcome this problem, such as resampling (García and Herrera, 2009), synthetic generation (Chawla et al., 2002) or penalized models, although they often are time consuming to implement (Zhou et al., 2017). iDT offers an easier way to overcome the problem by allowing the expert to force the tree to include the under-represented class by manually changing nodes' variable and thresholds to split and/or leaves nodes classes.

### *2.2. A python package and graphical user interface in jupyter lab for interactive construction and analysis of decision trees*

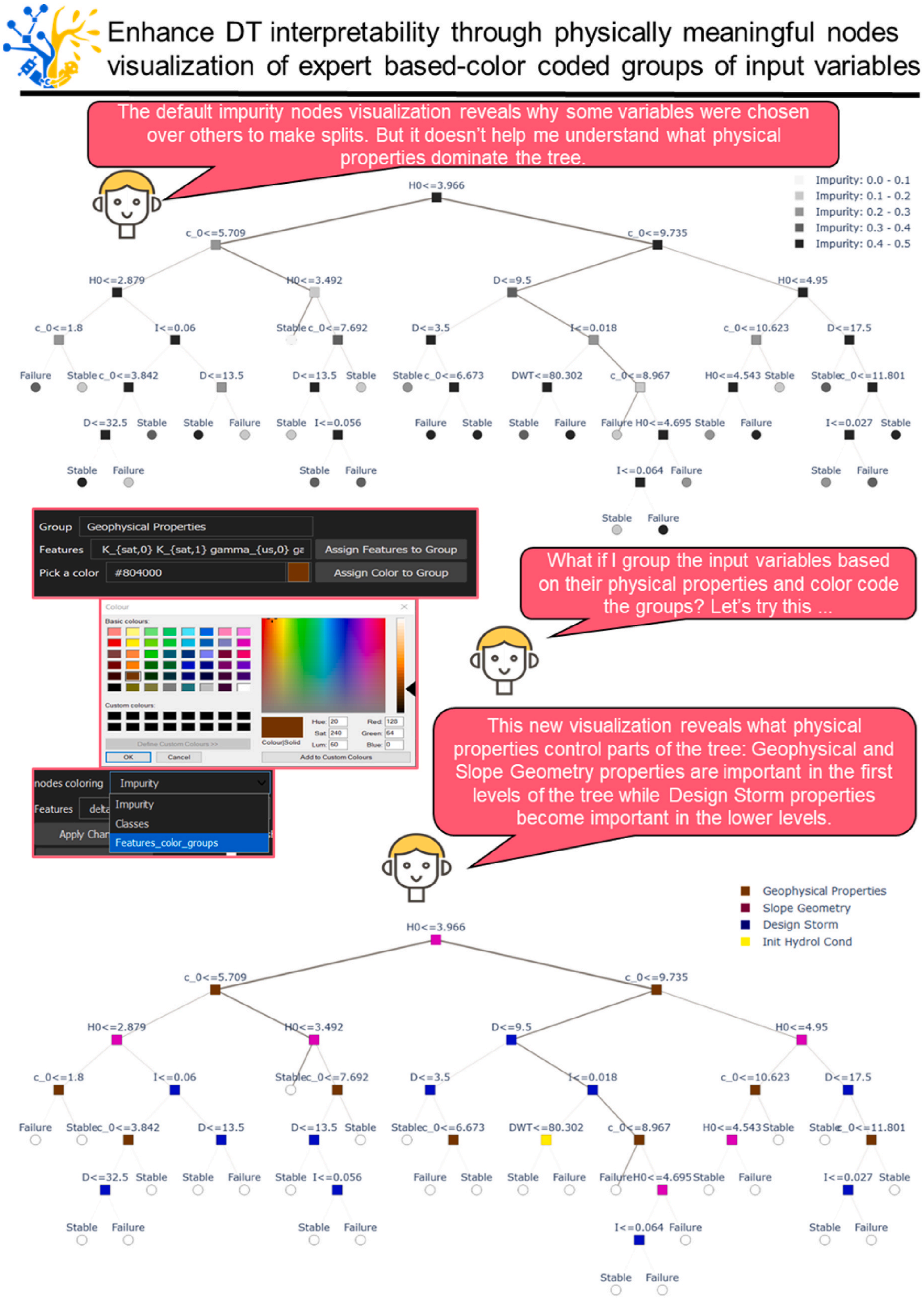
To maximise the reusability, replicability and reproducibility of our proposed approach (Gil et al., 2016; Hutton et al., 2016) we developed and shared an open-source Python package and a GUI in Jupyter Lab for implementing the IDTs framework. The code is available at.[3] We used the sklearn library of scikit-learn package in Python (Pedregosa et al., 2011) that contains the implementation of the tree algorithm (for more details see Supplementary material) as a basis for our interactive tools. We created a new package, called "InteractiveDT", which consists of (1) an "iDT" module containing the functions that enable the expert to interact with the DT or the dataset, and (2) an "iDTGUIfun" module which incorporates these functions into widgets, which are then used in the Jupyter Lab script called "InteractiveDecisionTrees" to create the user interface. Further details about this GUI are also provided in the Supplementary material.

### *2.3. Evaluating DT predictive and interpretive performance*

Decision trees are generally used as predictive tools for classification or regression. Therefore, their evaluation is typically based on statistical metrics of their predictive ability (Lipton, 2018). Examples of such metrics include classification accuracy, confusion matrices, precision, recall, accuracy rate, root mean squared error metrics, and mean error metrics (Pedregosa et al., 2011). However, in geosciences applications, we often would like the DT to be not just a good predictor, but also to be interpretable (Lipton, 2018). In contrast to predictive performance, interpretability is a less well-defined concept and metrics to measure interpretability are not yet well established (Doshi-Velez and Been, 2017). A widely used proxy for interpretability is the complexity of the tree, as it can be reasonably assumed that a less complex tree is easier to interpret (Molnar, 2020; Lipton, 2018). The complexity of a DT can be easily quantified through the number of leaf nodes and/or the depth of the tree (Molnar, 2020). We adopted these simple metrics to evaluate DT interpretability in our first case study.

The need for interpretability is often linked to the use of models to assist scientific understanding (Doshi-Velez and Been (2017)). The evaluation of interpretability for scientific understanding though is

[3] https://github.com/Sarailidis/Interactive-Decision-Trees.


Enhance DT interpretability through physically meaningful nodes visualization of expert based-color coded groups of input variables
The default impurity nodes visualization reveals why some variables were chosen over others to make splits. But it doesn't help me understand what physical properties dominate the tree.
Impurity: 0.0 - 0.1
Impurity: 0.1 - 0.2
Impurity: 0.2 - 0.3
Impurity: 0.3 - 0.4
Impurity: 0.4 - 0.5
H0<=3.966
c_0<=5.709
c_0<=9.735
H0<=2.879
H0<=3.492
D<=9.5
H0<=4.95
c_0<=1.8
I<=0.06
c_0<=7.692
D<=3.5
I<=0.018
c_0<=10.623
D<=17.5
c_0<=3.842
D<=13.5
c_0<=6.673
DWT<=80.302
c_0<=8.967
H0<=4.543
c_0<=11.801
D<=32.5
I<=0.056
H0<=4.695
I<=0.027
I<=0.064
Stable
Failure
What if I group the input variables based on their physical properties and color code the groups? Let's try this ...
This new visualization reveals what physical properties control parts of the tree: Geophysical and Slope Geometry properties are important in the first levels of the tree while Design Storm properties become important in the lower levels.
Geophysical Properties
Slope Geometry
Design Storm
Init Hydrol Cond

Group: Geophysical Properties

Features: K_{sat,0} K_{sat,1} gamma_{us,0} ga

Pick a color: #804000

Assign Features to Group
Assign Color to Group
Colour
Basic colours:
Custom colours:
Define Custom Colours >>
OK
Cancel
Colour|Solid
Add to Custom Colours

Hue: 20 Sat: 240 Lum: 60 Red: 128 Green: 64 Blue: 0

nodes coloring: Impurity

Features: delta

Impurity
Classes
Features_color_groups
Apply Char

**Fig. 3.** Decision Tree (top) for Case study 1 with the conventional nodes coloring approach, which is based on node impurity. The same tree is shown in the bottom with the proposed alternative nodes coloring, which is based on groups of variables proposed by the user. With this node coloring option, it is evident what kind of variables dominate the tree. The figure also shows a screenshot of the InteractiveDT tool developed to achieve this alternative visualization.

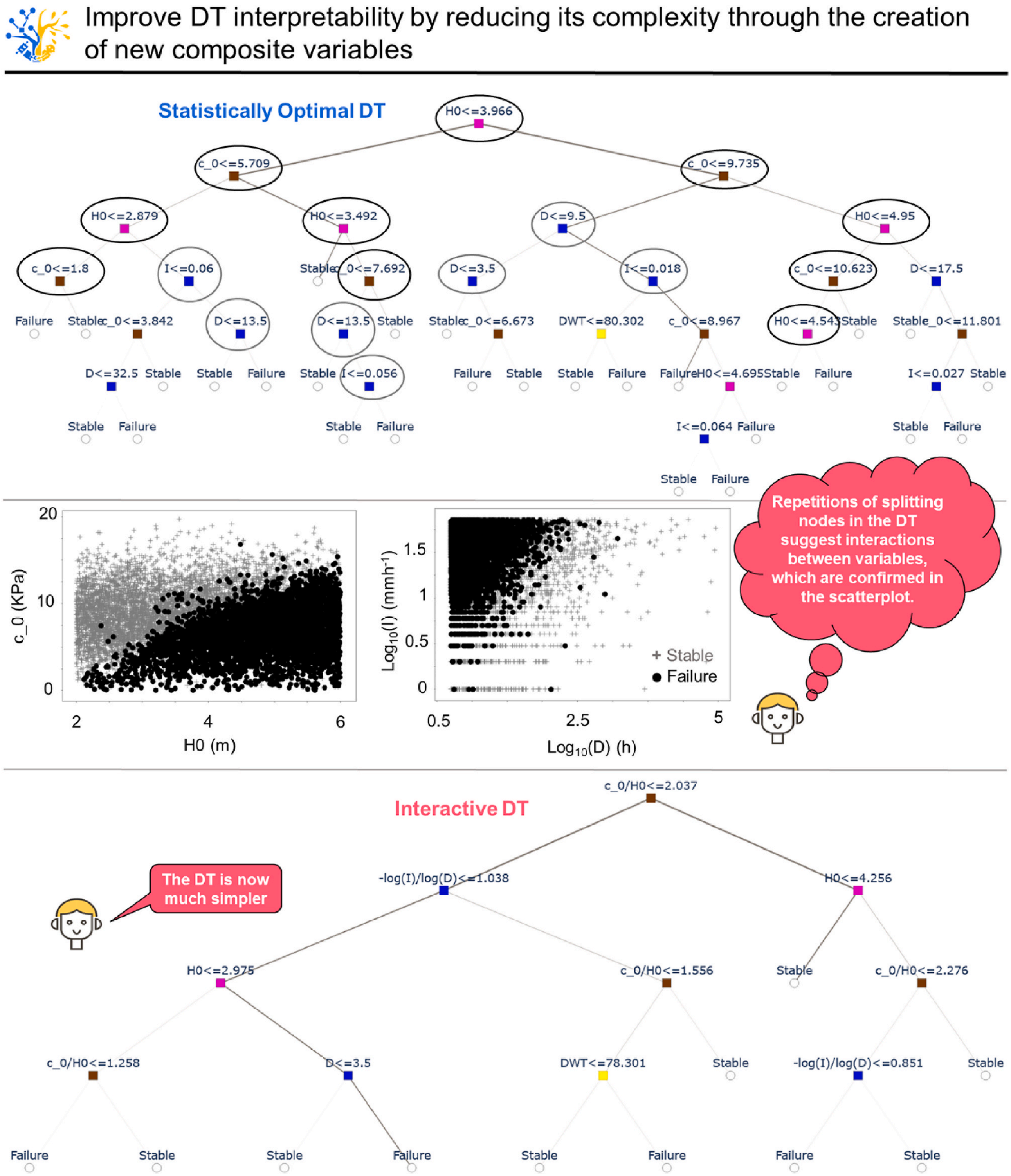


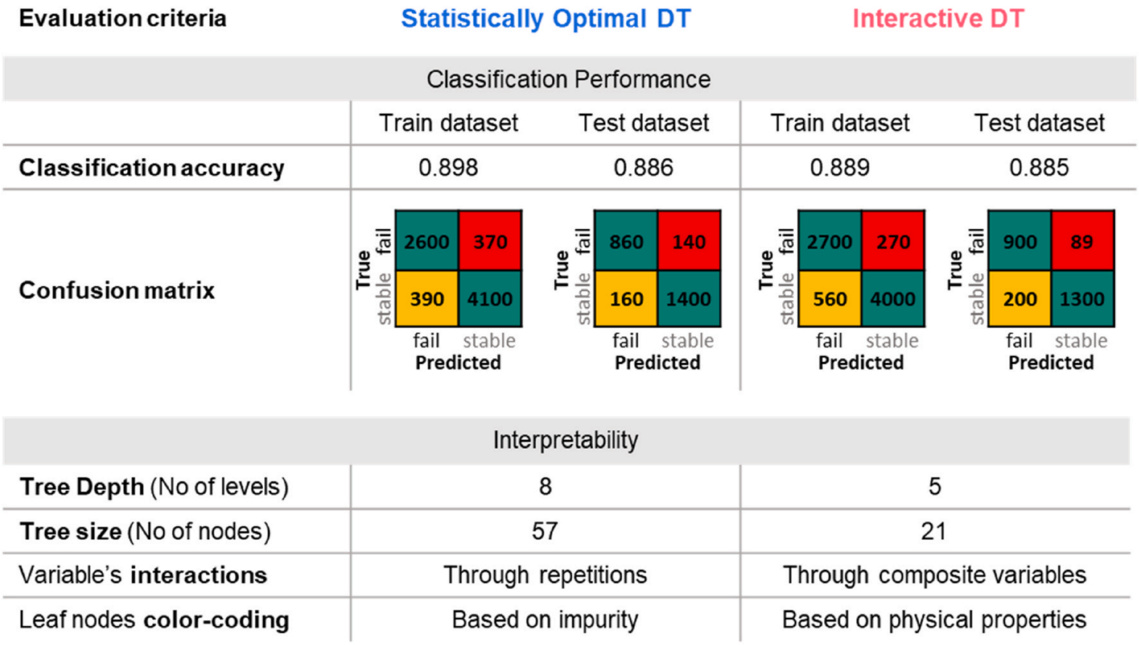
**Fig. 4.** Initial DT (top) for Case study 1, in which interactions between variables emerged. In the middle, the scatter plots of the interacting variables are shown, coloured according to whether the associated slope fails (black dots) or not (grey). The iDT in the bottom is the new tree after creating two composite variables based on the detected interactions.

context specific. In our second case study, we will give an example of a case-specific definition of interpretability, based on the consistency of the DT partitioning with an existing independent classification system of some of the input variables.

## 3. Results

### 3.1. *Case study 1 – color-coding groups of variables and constructing new composite variables to reduce the DT complexity and increase interpretability*

The first case study is based on a dataset from a computational



| Evaluation criteria | Statistically Optimal DT | | Interactive DT | |
|---|---|---|---|---|
| Classification Performance | | | | |
| | Train dataset | Test dataset | Train dataset | Test dataset |
| **Classification accuracy** | 0.898 | 0.886 | 0.889 | 0.885 |
| **Confusion matrix** | True fail: 2600, 370<br>True stable: 390, 4100<br>Predicted: fail, stable | True fail: 860, 140<br>True stable: 160, 1400<br>Predicted: fail, stable | True fail: 2700, 270<br>True stable: 560, 4000<br>Predicted: fail, stable | True fail: 900, 89<br>True stable: 200, 1300<br>Predicted: fail, stable |
| Interpretability | | | | |
| **Tree Depth** (No of levels) | 8 | | 5 | |
| **Tree size** (No of nodes) | 57 | | 21 | |
| Variable's **interactions** | Through repetitions | | Through composite variables | |
| Leaf nodes **color-coding** | Based on impurity | | Based on physical properties | |

**Fig. 5.** Evaluation and comparison of the statistically optimal and interactive DTs for Case Study 1 based on classification performance and interpretability.

landslide study by Almeida et al. (2017), which includes 10,000 combinations of 28 input variables of a slope stability model (the list is given in Table S1 in the supplementary material). These variables are model inputs characterising landscape attributes such as slope geometry, soil and design storm properties and initial hydrological conditions. The model output is the slope factor of safety (FoS), which is typically used to separate the model outputs into two results regarding the stability of a landscape position regarding landslide hazard risk: "stable", when FoS is above 1, and "failure" otherwise. In Almeida et al. (2017) a standard CART algorithm was used to identify dominant drivers of slope instability. We applied our iDT procedure to the same dataset to demonstrate two functionalities of our iDT toolbox: (a) How to increase the visual interpretability of the DT by colour coding variables based on their physical meaning. (b) How to better capture interactions between variables by creating new composite variables.

Fig. 3 shows the statistically optimal DT delivered by the automatic DT algorithm. Nodes are coloured based on Impurity, a default choice in many software packages. Fig. 3 also shows the graphical interface of the InteractiveDT tool, which enables the user to define groups of input variables and colour code the nodes accordingly. Through this new visualization, it is evident that the first three levels of the tree are dominated by "geophysical properties" and "slope geometry variables", while levels 4 and 5 are mainly dominated by "design storm properties". Furthermore, the colour coding helps spotting a repetition of four variables - cohesion (c_0), thickness of topsoil (H0), rainfall intensity (I) and duration (D) - at various levels of the tree. Such a repetition suggests that these variables may be interacting with one another to produce slope failures (the tree tries to mimic this interaction). Indeed, a scatterplot of c_0 versus H0 (left hand side in Fig. 4) shows that combinations of high H0 and low c_o (bottom right) lead to slope failure (black dots). Moreover, we expect rainfall intensity and duration to interact. Specifically, combinations of high-intensity/short-duration and low-intensity/long-duration rainfall will more likely result in slope failure. This relationship is confirmed in the (log-scale) scatterplot on the right-hand side of Fig. 4. To capture these interactions, the user may create two composite variables: Soil Ratio, which is the ratio of cohesion and thickness of topsoil (Soil Ratio = c_0/H0), and Storm Ratio, which is the ratio of the logarithms of rainfall intensity and duration (Storm Ratio = -log10(D)/log10(I)). The bottom part of Fig. 4 shows the new DT delivered by the algorithm when fed by a training dataset including these two composite variables. Overall, the new DT is "better" than the original one because it is smaller (21 nodes instead of 57, and a depth of 5 layers instead of 8) and hence easier to interpret, and it is more accurate in predicting slope failure (higher number of true slope failures and lower number of false slope stabilities) for both training and test datasets. Evaluation and comparison of the two DTs are summarised in Fig. 5.

### 3.2. *Case study 2 – Increasing interpretability by changing splitting threshold values based on other relevant knowledge sources*

The second case study is based on a version of groundwater recharge dataset created by Sarrazin (2018) which includes 17,000,000 simulations of 34 input variables of a hydrological model. These variables are model inputs characterizing spatially distributed climate properties, land cover and soil properties of karst landscapes across Europe under current conditions and future climate scenarios. The model outputs are values of annual groundwater recharge, which are then grouped into four classes, namely, C1 (<20 mm/yr), C2 (20–100 mm/yr), C3 (100–300 mm/yr) and C4 (>300 mm/yr). A DT is built to reveal the key controls of groundwater recharge. To increase the interpretability of the DT we used our iDT framework to manually change some of the nodes' thresholds consistently with a simplified version of Holdridge's life zones classification scheme (Holdridge, 1947). The Holdridge scheme provides a classification of land areas based on annual precipitation and aridity index (i.e. the ratio between potential evaporation and precipitation; Figure S2 in the Supplementary material shows the original and our simplified scheme). By imposing that the threshold values for Precipitation (Pm) and Aridity index (AI) in the DT be the same as in the Holdridge chart thresholds, we wanted to explore whether a tree so constructed leads to leaf nodes that map into fewer Holdridge life zones, and as such may be more interpretable, and whether this gain in interpretability comes with a significant loss in classification accuracy.

We generated 15 datasets of 1000 samples each by randomly sampling from the original dataset (of 17,000,000 samples). For each dataset we derived a statistically optimal (SO) and an interactive (iDT) decision tree. To derive the SO decision tree, we tried different combinations of the algorithm tuning parameters (splitting criterion based on "Gini impurity" or "entropy", maximum number of leaf nodes varied from 15 to 25, maximum impurity decreases of $10^{-5}$, $10^{-6}$, $10^{-7}$) and retained the best SO tree based on 10-fold Cross Validation strategy. To derive the

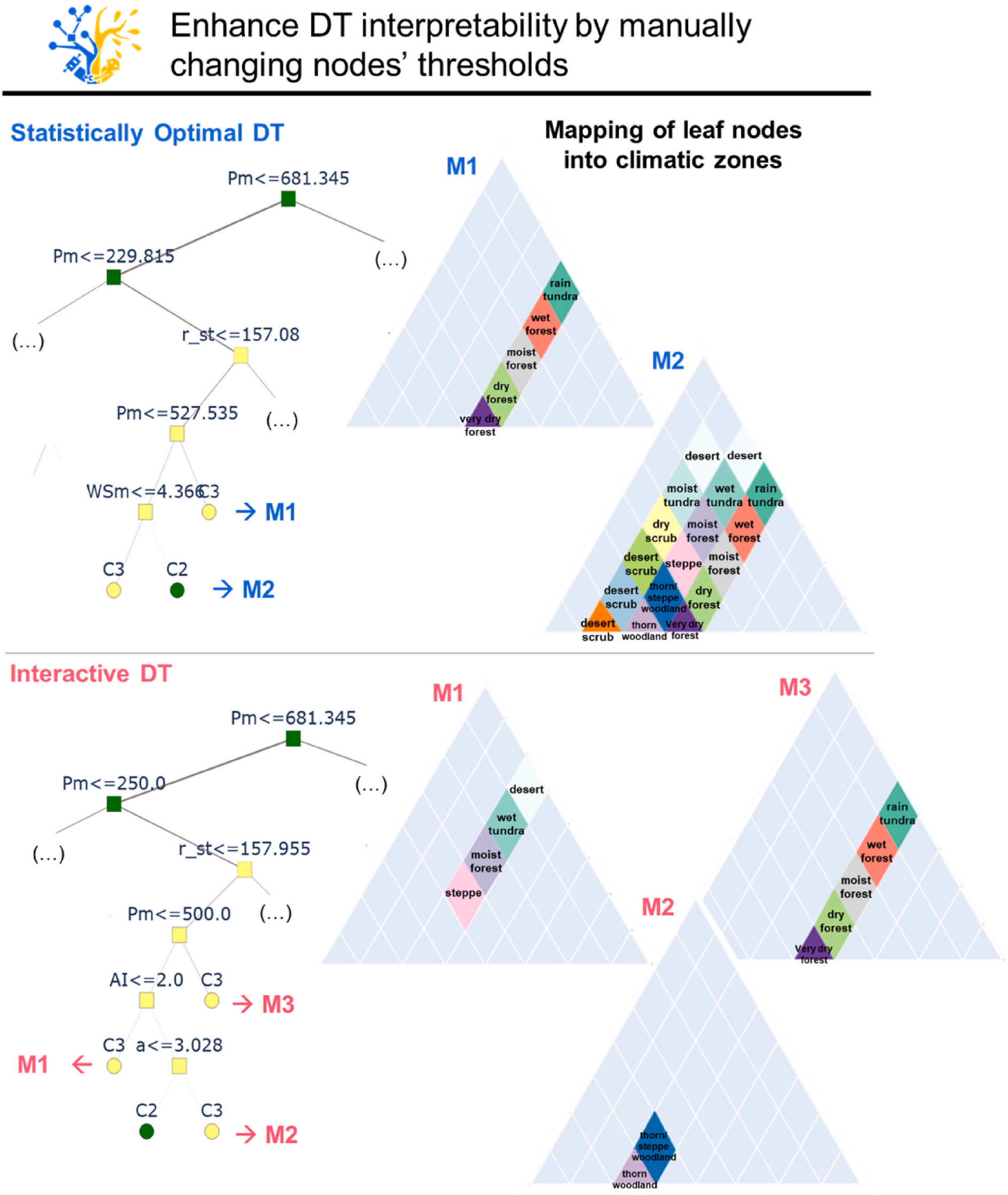


**Fig. 6.** Detail of the statistically optimal DT (top) for Case Study 2 and of the iDT (bottom) after manually changing the thresholds for Precipitation (Pm) and Aridity index (AI). For the leaves nodes of each DT we plotted the Holdridge scheme and highlighted the diamonds that the leaves can be mapped to.

corresponding iDT, we used the iDT framework to manually change all the splitting thresholds for Pm and AI to the closest Holdridge chart threshold values. The closest threshold could sometimes be quite far, and the choice of changing it, is subjective. But in some other cases even a small change e.g., changing AI threshold from 1.1 to 1, could make a big difference in terms of interpretability.

Fig. 6 shows an example of a statistically optimal DT (top) and the corresponding iDT (bottom), focusing on the specific branch where we manually changed the thresholds for Precipitation (Pm) and Aridity index (AI), and hence the resulting leaf nodes. Next to the tree branches, we show the Holdridge life zones (HLZs) that the leaf nodes are mapped into. In the statistically optimal DT (top), the three leaf nodes map into 13 and 5 HLZs respectively. In the iDT (bottom), the number of HLZs is reduced to 4, 2 and 5 after changing one precipitation threshold from 229.82 to 250 and another one from 527.54 to 500 (in line with the HLZ classification). Of particular interest is the leaf nodes labelled C2, which define conditions under which groundwater recharge is low. In the statistically optimal tree, such conditions appear in 13 different climatic zones, while in the iDT they can only appear in two climates: thorn/ steppe or thorn woodland. This drastic reduction opens up the

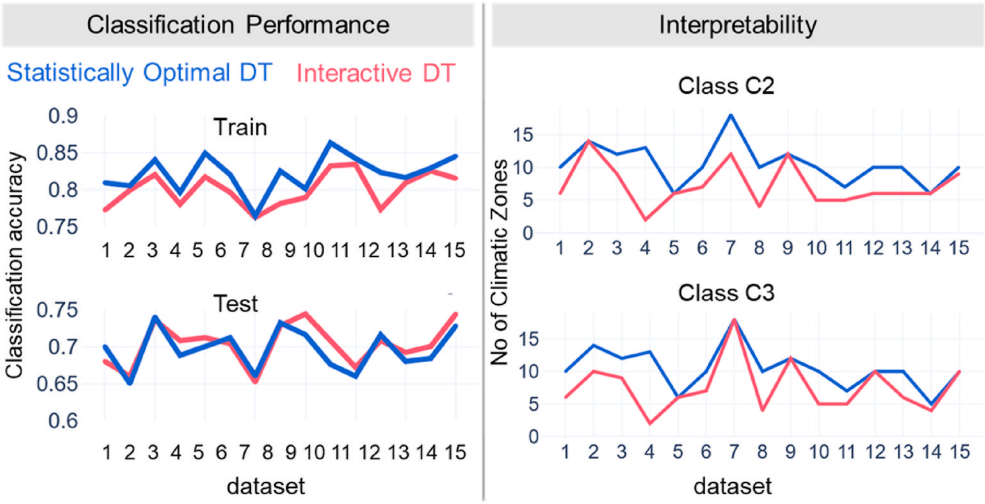


**Fig. 7.** Evaluation and comparison of the statistically optimal and interactive DTs for Case Study 2 based on their classification performance and interpretability.

opportunity for the expert to find meaningful explanations of why those two particular climatic zones exhibit lower recharge and the implications of this finding. While it is beyond the scope of this paper to go into such explanation, our argument is that the opportunity to develop it in the first place would have not been present if using the statistically optimal tree. Since class C2 was associated to a variety of different climatic zones.

Fig. 7 shows the classification and interpretability performance of all 15 statistically optimal DTs (one per each of the 15 datasets subsamples) and associated iDTs (each obtained from the statistically optimal DT by changing Pm and AI thresholds). Regarding classification performance, the differences are not pronounced, which means the changes made by the expert did not lead to a significant loss of performance. As expected, the statistically optimal DTs always show a slightly higher classification accuracy in the training sets. Interestingly though, the iDTs outperform the statistically optimal trees in most cases (9 out of 15) in the test sets. Interpretability performance was quantified through the number in climatic zones classes C2 and C3 classes can be mapped to. Overall, the plot shows that the number of HLZs associated to leaf nodes of classes C2 and C3 tends to decrease. In conclusion, this example shows that incorporating other knowledge sources in the DT development by manually changing the splitting thresholds produces iDTs with a clearer link to that knowledge, and hence higher interpretability potential, at no significant loss in classification accuracy.

### 3.3. *Case study 3 manually changing nodes' variables and threshold values to include under-represented classes in imbalanced datasets*

This case study is an example of application of iDT in cases where certain classes are under-represented in a dataset, a situation known as "imbalanced datasets". We again used the dataset from Sarrazin (2018) as in Sec. 3.2, and randomly generated 5 subsample datasets of increasing sizes (1000, 5000, 10000, 50000 and 100000 samples). We then split each subsample dataset into a training and a test set (75% and 25% of the dataset size respectively) and randomly removed data points that belonged to class C2 from the training dataset. Therefore, the training sets contained only few data points of class C2 ($<2\%$). Similarly, to Sec. 3.2, for each dataset we trained a Statistically Optimal (SO) decision tree and then derived an iDT by manually changing the nodes' variables and thresholds until the iDT included the unrepresented class C2 in some of its leaf nodes. In some cases, we also manually changed the class of a leaf node to class C2. For example, in Fig. 8 on the left we show a part of the SO tree obtained for sample dataset 2. We know from Sarrazin (2018) that low recharge class C2 should appear for low precipitation values, but the algorithm fails to include the C2 class in the SO tree as the class is under-represented in the training dataset. Hence, based on the splitting variables and thresholds of the DT found in Sarrazin (2018) we manually changed the threshold in the split node *"Pm<=639.075"* to *"Pm<=300"* and the node variable in the split *"Vr<=201.14"* to *"Pm<=65"* in our DT, so to create a branch in the tree that specifically explore low precipitation cases. In response to these manual changes, the algorithm created a leaf node for class C2 in the iDT (top right of Fig. 8). The change induced a loss of classification accuracy in the training dataset (see Fig. 8, case '2') but an increase in performance on the test dataset against unseen data. Moreover, the confusion matrices indicate that iDT is more capable in correctly classifying data points into Class C2 as shown in Fig. S3 in the Supplementary material. A similar trend is found for all other datasets: as expected, SO trees perform better in the training sets but iDTs outperform SO trees in in test set, particularly for smaller datasets.

## 4. Conclusions

How we can incorporate scientific knowledge into ML models to improve their physical realism, their robustness and their interpretability remains a major challenge and opportunity for ML applications in the geosciences (Read et al., 2019; Sun et al., 2022). To address this problem, we propose a framework for the construction and analysis of interactive decision trees (iDTs) for application in the geosciences. We created an open-source implementation of iDT in Python and Jupyter Lab, which we hope will encourage the use of iDT in future research applications. We demonstrated the iDT approach in three case studies that represent typical challenges encountered in applications of decisions trees in the geosciences. We found that our proposed iDT framework supports the development of decision trees that are easier to visualise and interpret in a physical sense. In our second case study, we find that manual adjustment of splitting thresholds can lead to a more physically meaningful tree with almost no loss in classification performance. In the third example, we show how experts can build a more robust and physically consistent DT in cases of imbalanced datasets that can generalize better on unseen data. Even though manually changing the nodes' variables and threshold values based on domain knowledge to consider an under-represented class deteriorated the classification accuracy in training sets, it improved it in test sets.

An important direction for future geoscience research is to achieve closer interaction between human experts and machine learning algorithms by including domain knowledge in algorithmic form (Solomatine and Ostfeld, 2008). For example, experts could force the algorithm to search for thresholds in a specific range of values for selected variables, or they could define constraints on variable selection to eliminate

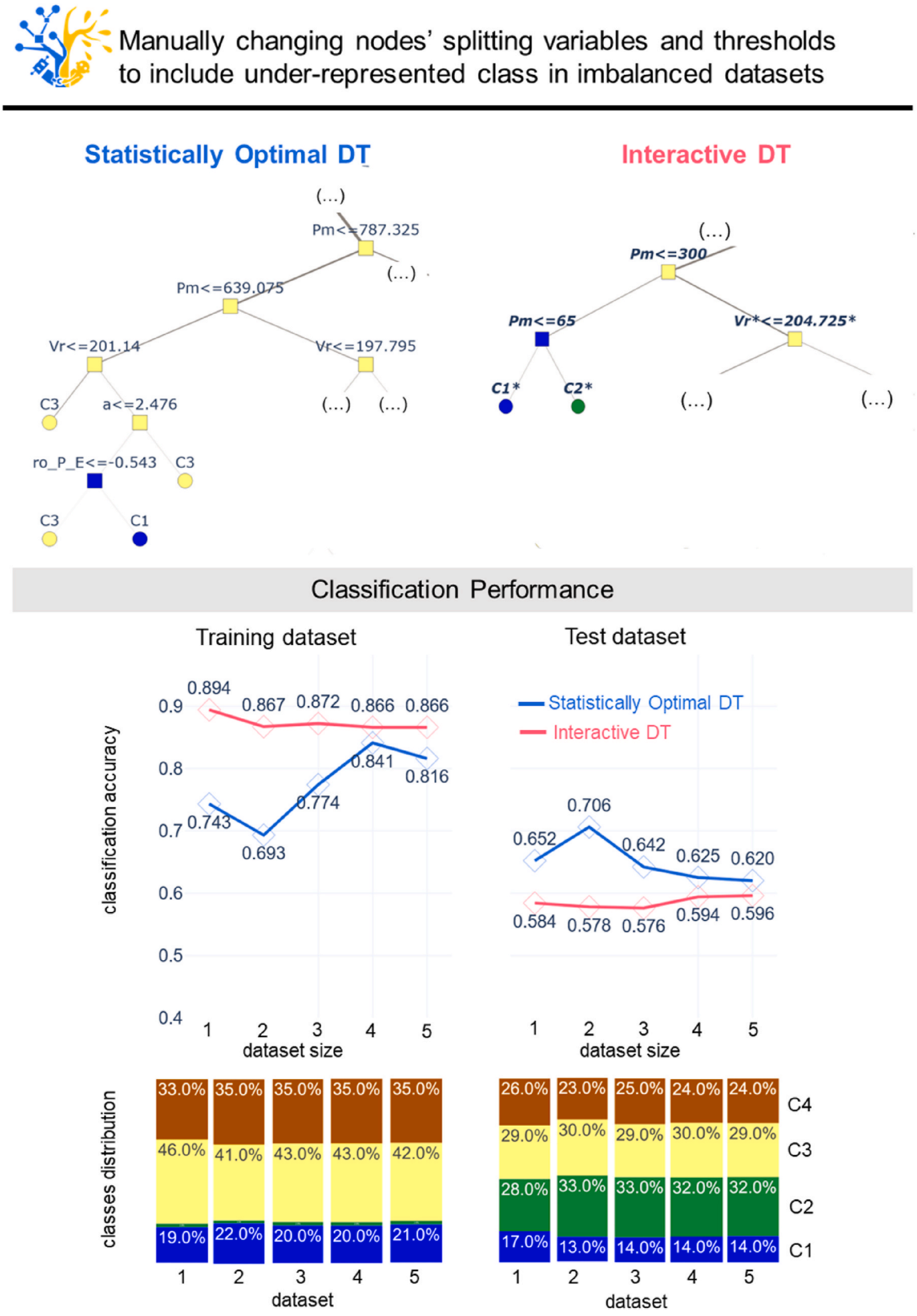


**Fig. 8.** Detail of the statistical optimal DT (top left) for Case Study 3 and interactive DT (right) for the sample dataset 2. In the iDT, the variables and threshold values in italic are those manually changed by the user, and those marked with an asterisk are changed by the DT algorithm in response to the manual changes. The bottom panel shows the classification accuracies on the training and test sets, and the distribution of the four output classes (C1–C4), in each of the 5 datasets (bottom).

unrealistic sequence of variables to split. Another area for future improvement would be to expand the range of visualization techniques (e.g. partial dependence plots, accumulated local effects, feature interaction; see for example application in Shortridge et al. (2016)) that could be used in parallel to further enhance DT interpretability.

We hope that this paper will contribute to foster the development and use of interactive decision trees and, more broadly, of methods to better integrate domain knowledge in ML, which is particularly relevant for geoscience applications.

## Authorship contribution statement

Georgios Sarailidis: Georgios Sarailidis developed the proposed method & toolbox and prepared the manuscript; Thorsten Wagener: Thorsten Wagener supervised the method development and testing and revised the manuscript. Francesca Pianosi: Francesca Pianosi supervised the method development and testing and revised the manuscript.

## Code availability section

Name of the code/library: InteractiveDT, (GPL-3.0 License) (version v1.1.0).

Contact: g.sarailidis@bristol.ac.uk, 00447957332324.

Hardware requirements: The presented toolbox has been tested on a computer with the following characteristics:

- Processor: Intel(R) Core (TM) i7-8700 CPU @ 3.20 GHz 3.19 GHz
- RAM: 16.0 GB (15.8 GB useable)
- System Type: 64-bit operating system, x64-based processor

Program language: Python.

Software required: python, jupyter lab, anaconda navigator.

Program size: 4658 KB.

Access to the code, datasets and workflows to reproduce the results presented in this paper: https://github.com/Sarailidis/Interactive-Decision-Trees.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Links for the data/code are provided in the manuscript and in the code availability section

## Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council in the UK via grant [grant number EP/L016214/1] awarded for the Water Informatics: Science and Engineering (WISE) Centre for Doctoral Training, which is gratefully acknowledged. Francesca Pianosi is partially supported by the Engineering and Physical Sciences Research Council through an Early Career “Living with Environmental Uncertainty” Fellowship [grant number EP/R007330/1]. Support for Thorsten Wagener was provided by the Alexander von Humboldt Foundation in the framework of the Alexander von Humboldt Professorship endowed by the German Federal Ministry of Education and Research.

## Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.cageo.2022.105248.

## References

Addor, N., Nearing, G., Prieto, C., Newman, A.J., le Vine, N., Clark, M.P., 2018. A ranking of hydrological signatures based on their predictability in space. Water Resour. Res. 54, 8792–8812. https://doi.org/10.1029/2018WR022606.

Almeida, S., Ann Holcombe, E., Pianosi, F., Wagener, T., 2017. Dealing with deep uncertainties in landslide modelling for disaster risk reduction under climate change. Nat. Hazards Earth Syst. Sci. 17, 225–241. https://doi.org/10.5194/nhess-17-225-2017.

Ankerst, M., Ester, M., Kriegel, H.P., 2000. Towards an effective cooperation of the user and the computer for classification. In: Proceeding of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 179–188. https://doi.org/10.1145/347090.347124.

Bergen, K.J., Johnson, P.A., de Hoop, M.v., Beroza, G.C., 2019. Machine learning for data-driven discovery in solid Earth geoscience. Science 363. https://doi.org/10.1126/science.aau0323.

Beven, K.J., Almeida, S., Aspinall, W.P., Bates, P.D., Blazkova, S., Borgomeo, E., Freer, J., Goda, K., Hall, J.W., Phillips, J.C., Simpson, M., Smith, P.J., Stephenson, D.B., Wagener, T., Watson, M., Wilkins, K.L., 2018. Epistemic uncertainties and natural hazard risk assessment - Part 1: a review of different natural hazard areas. Nat. Hazards Earth Syst. Sci. 18, 2741–2768. https://doi.org/10.5194/nhess-18-2741-2018.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees, Classification and Regression Trees. CRC Press. https://doi.org/10.1201/9781315139470.

Butler, D., 2007. Earth monitoring: the planetary panopticon. Nature 450, 778–781. https://doi.org/10.1038/450778a.

Bzdok, D., Krzywinski, M., Altman, N., 2017. Machine learning: a primer. Nat. Methods 14, 1119–1120. https://doi.org/10.1038/nmeth.4526.

Chawla, N.v., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357. https://doi.org/10.1613/jair.953.

Do, T., 2006. Towards simple, easy-to-understand, an interactive decision tree algorithm. In: 9th National Conference in Computer Science.

Doshi-Velez, F., Been, K., 2017. Towards A Rigorous Science of Interpretable Machine Learning.

Elia, M., Gajek, C., Schiendorfer, A., Wolfgang, R., 2021. An interactive web application for decision tree learning. In: Proceedings of the European Conference on Machine Learning, Teaching Machine Learning Workshop. Ghent, Belgium.

Estivill-Castro, V., Gilmore, E., Hexel, R., 2020. Human-in-the-loop construction of decision tree classifiers with parallel coordinates. In: Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics. https://doi.org/10.1109/SMC42975.2020.9283240.

Faghmous, J.H., Kumar, V., 2014. A big data guide to understanding climate change: the case for theory-guided data science. Big Data 2, 155–163. https://doi.org/10.1089/big.2014.0026.

Fails, J.A., Olsen, D.R., 2003. Interactive machine learning. In: International Conference on Intelligent User Interfaces. Proceedings IUI. https://doi.org/10.1145/604045.604056.

Flach, P., 2012. Machine Learning the Art and Science of Algorithms that Make Sense of Data. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511973000.

García, S., Herrera, F., 2009. Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. Evol. Comput. 17, 275–306. https://doi.org/10.1162/evco.2009.17.3.275.

Gil, Y., David, C.H., Demir, I., Essawy, B.T., Fulweiler, R.W., Goodall, J.L., Karlstrom, L., Lee, H., Mills, H.J., Oh, J.H., Pierce, S.A., Pope, A., Tzeng, M.W., Villamizar, S.R., Yu, X., 2016. Toward the Geoscience Paper of the Future: best practices for documenting and sharing research from data to software to provenance. Earth Space Sci. https://doi.org/10.1002/2015EA000136.

Gislason, P.O., Benediktsson, J.A., Sveinsson, J.R., 2006. Random forests for land cover classification. Pattern Recogn. Lett. 27, 294–300. https://doi.org/10.1016/j.patrec.2005.08.011.

Grimm, R., Behrens, T., Märker, M., Elsenbeer, H., 2008. Soil organic carbon concentrations and stocks on Barro Colorado Island - digital soil mapping using Random Forests analysis. Geoderma 146, 102–113. https://doi.org/10.1016/j.geoderma.2008.05.008.

Han, J., Cercone, N., 2001. Interactive construction of decision trees. In: 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1007/3-540-45357-1_61.

Hart, J.K., Martinez, K., 2006. Environmental Sensor Networks: a revolution in the earth system science? Earth Sci. Rev. 78, 177–191. https://doi.org/10.1016/j.earscirev.2006.05.001.

Hengl, T., de Jesus, J.M., Heuvelink, G.B.M., Gonzalez, M.R., Kilibarda, M., Blagotić, A., Shangguan, W., Wright, M.N., Geng, X., Bauer-Marschallinger, B., Guevara, M.A., Vargas, R., MacMillan, R.A., Batjes, N.H., Leenaars, J.G.B., Ribeiro, E., Wheeler, I., Mantel, S., Kempen, B., 2017. SoilGrids250m: global gridded soil information based on machine learning. PLoS One 12. https://doi.org/10.1371/journal.pone.0169748.

Holdridge, L.R., 1947. Determination of world plant formations from simple climatic data. Science 105. https://doi.org/10.1126/science.105.2727.367, 1979.

Hutton, C., Wagener, T., Freer, J., Han, D., Duffy, C., Arheimer, B., 2016. Most computational hydrology is not reproducible, so is it really science? Water Resour. Res. https://doi.org/10.1002/2016WR019285.

IBM, 2020. Machine learning [WWW document], 6.16.21. https://www.ibm.com/cloud/learn/machine-learning.

Iorgulescu, I., Beven, K.J., 2004. Nonparametric direct mapping of rainfall-runoff relationships: an alternative approach to data analysis and modeling? Water Resour. Res. 40 https://doi.org/10.1029/2004WR003094.

Karpatne, A., Atluri, G., Faghmous, J.H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., Kumar, V., 2017. Theory-guided data science: a new paradigm for scientific discovery from data. IEEE Trans. Knowl. Data Eng. 29, 2318–2331. https://doi.org/10.1109/TKDE.2017.2720168.

Karpatne, A., Ebert-Uphoff, I., Ravela, S., Babaie, H.A., Kumar, V., 2019. Machine learning for the geosciences: challenges and opportunities. IEEE Trans. Knowl. Data Eng. 31, 1544–1554. https://doi.org/10.1109/TKDE.2018.2861006.

Kirchner, J.W., Berghuijs, W.R., Allen, S.T., Hrachowitz, M., Hut, R., Rizzo, D.M., 2020. Streamflow response to forest management. Nature 578, E12–E15. https://doi.org/10.1038/s41586-020-1940-6.

Kuentz, A., Arheimer, B., Hundecha, Y., Wagener, T., 2017. Understanding hydrologic variability across Europe through catchment classification. Hydrol. Earth Syst. Sci. 21, 2863–2879. https://doi.org/10.5194/hess-21-2863-2017.

Lipton, Z.C., 2018. The mythos of model interpretability. Commun. ACM 61, 36–43. https://doi.org/10.1145/3233231.

Loh, W.Y., 2014. Fifty years of classification and regression trees. Int. Stat. Rev. 82, 329–348. https://doi.org/10.1111/insr.12016.

Mickens, J., Szummer, M., Narayanan, D., 2007. Snitch: interactive decision trees for troubleshooting misconfigurations. In: 2nd Workshop on Tackling Computer Systems Problems with Machine Learning Techniques. SysML 2007, Co-Located with NSDI 2007.

Molnar, C., 2020. Interpretable Machine Learning. A Guide for Making Black Box Models Explainable. Book.

Pal, M., Mather, P.M., 2003. An assessment of the effectiveness of decision tree methods for land cover classification. Remote Sens. Environ. 86, 554–565. https://doi.org/10.1016/S0034-4257(03)00132-9.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Read, J.S., Jia, X., Willard, J., Appling, A.P., Zwart, J.A., Oliver, S.K., Karpatne, A., Hansen, G.J.A., Hanson, P.C., Watkins, W., Steinbach, M., Kumar, V., 2019. Process-Guided deep learning predictions of lake water temperature. Water Resour. Res. 55, 9173–9190. https://doi.org/10.1029/2019WR024922.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., Prabhat, 2019. Deep learning and process understanding for data-driven Earth system science. Nature 566, 195–204. https://doi.org/10.1038/s41586-019-0912-1.

Roscher, R., Bohn, B., Duarte, M.F., Garcke, J., 2020. Explainable machine learning for scientific insights and discoveries. IEEE Access 8, 42200–42216. https://doi.org/10.1109/ACCESS.2020.2976199.

Samuel, A.L., 1959. Some studies in machine learning using the game of checkers. IBM J. Res. Dev. 3, 210–229. https://doi.org/10.1147/rd.33.0210.

Sarrazin, F., 2018. Understanding the Sensitivity of Karst Groundwater Recharge to Climate and Land Cover Changes at a Large-Scale. PhD Dissertation. University of Bristol, Bristol, United Kingdom.

Sawicz, K.A., Kelleher, C., Wagener, T., Troch, P., Sivapalan, M., Carrillo, G., 2014. Characterizing hydrologic change through catchment classification. Hydrol. Earth Syst. Sci. 18, 273–285. https://doi.org/10.5194/hess-18-273-2014.

Shortridge, J.E., Guikema, S.D., Zaitchik, B.F., 2016. Machine learning methods for empirical streamflow simulation: a comparison of model accuracy, interpretability, and uncertainty in seasonal watersheds. Hydrol. Earth Syst. Sci. 20 https://doi.org/10.5194/hess-20-2611-2016.

Singh, R., Archfield, S.A., Wagener, T., 2014. Identifying dominant controls on hydrologic parameter transfer from gauged to ungauged catchments - a comparative hydrology approach. J. Hydrol. 517, 985–996. https://doi.org/10.1016/j.jhydrol.2014.06.030.

Solomatine, D.P., Ostfeld, A., 2008. Data-driven modelling: some past experiences and new approaches. J. Hydroinf. https://doi.org/10.2166/hydro.2008.015.

Solomatine, D.P., Siek, M.B.L.A., 2004. Flexible and optimal M5 model trees with applications to flow predictions. In: Hydroinformatics. https://doi.org/10.1142/9789812702838_0212.

Stein, L., Pianosi, F., Woods, R., 2020. Event-based classification for global study of river flood generating processes. Hydrol. Process. 34 https://doi.org/10.1002/hyp.13678.

Sun, Z., Sandoval, L., Crystal-Ornelas, R., Mousavi, S.M., Wang, Jinbo, Lin, C., Cristea, N., Tong, D., Hawley Carande, W., Ma, X., Rao, Y., Bednar, J, A., Tan, A., Wang, Jianwu, Purushotham, S., Gill T, E., Chastang, J., Howard, D., Holt, B., Gangodagamage, C., Zhao, P., Rivas, P., Chester, Z., Orduz, J., John, A., 2022. A review of earth artificial intelligence. Comput. Geosci. 159 https://doi.org/10.1016/j.cageo.2022.105034.

Teoh, S.T., Ma, K.L., 2003. PaintingClass: interactive construction, visualization and exploration of decision trees. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 667–672. https://doi.org/10.1145/956750.956837.

van den Elzen, S., van Wijk, J.J., 2011. BaobabView: interactive construction and analysis of decision trees. In: VAST 2011 - IEEE Conference on Visual Analytics Science and Technology 2011, Proceedings, pp. 151–160. https://doi.org/10.1109/VAST.2011.6102453.

Washington, W.M., Buja, L., Craig, A., 2009. The computational future for climate and Earth system models: on the path to petaflop and beyond. Phil. Trans. Math. Phys. Eng. Sci. 367, 833–846. https://doi.org/10.1098/rsta.2008.0219.

Zhou, L., Pan, S., Wang, J., Vasilakos, A.v., 2017. Machine learning on big data: opportunities and challenges. Neurocomputing 237, 350–361. https://doi.org/10.1016/j.neucom.2017.01.026.