Interpretable AI/ML for High-stakes Tasks with Human-in-the-loop: Critical Review and Future Trends

Boris Kovalerchuk

borisk@cwu.edu

Central Washington University

Research Article

Keywords: Machine Learning, Interpretable AI/ML, Explainable AI/ML, High-stakes tasks, Human-in-the-loop, Trustworthy models, Visual Knowledge Discovery, LIME, SHAP, Explanations, Quasi-explanation

Posted Date: February 29th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-3989807/v1

Additional Declarations: No competing interests reported.

Interpretable AI/ML for High-stakes Tasks with Human-in-the-loop: Critical Review and Future Trends

Boris Kovalerchuk
Dept. of Computer Science,
Central Washington University, USA
borisk@cwu.edu

Abstract. Domains with high-stakes tasks include medical diagnosis, financial decision-making, autonomous vehicles, criminal justice, disaster response, search and rescue operations, and others. In high-stakes tasks the effects of incorrect decisions or predictions can cause significant harm to individuals or society. This paper presents (1) major topics of research in interpretable AI/ML models for high-stakes tasks with human-in-the-loop, and (2) motivates and explains topics of emerging importance in this area. It is widely accepted that model explanations should accurately describe the model's inner decision making and be convincing to the user. However, quite often these properties are missing. revealing only a quasi-explanation of the model, thus not serving the goal of true model explanation. This paper presents both benefits and deficiencies of current methods and ways to overcome deficiencies with a focus on high-stakes AI/ML tasks, where incorrect decisions or predictions can result in devastating consequences. While a human-in-the-loop is necessary for solving explainability problems, the success heavily depends on human cognitive abilities and limitations. This makes human visual knowledge discovery with granular computing critical for the progress of interpretable machine learning. Therefore, a significant part of the paper is devoted to presenting interpretable ML models built with human visual knowledge discovery methods.

Keywords: Machine Learning, Interpretable AI/ML, Explainable AI/ML, High-stakes tasks, Human-in-the-loop, Trustworthy models, Visual Knowledge Discovery, LIME, SHAP, Explanations, Quasi-explanation.

1. Introduction

1.1. Overview

A large variety of problem domains include high-stakes tasks, which require explanation of Artificial Intelligence/Machine Learning (AI/ML) models. Typical examples are *medical diagnosis* (cancer, heart diseases), *financial decision-making* (credit risk, fraud detection), *autonomous vehicles* (traffic prediction and pedestrian detection), *criminal justice* (predictive policing and sentencing recommendations), *disaster response planning* (identify areas at risk of disaster – hurricanes, floods, and earthquakes), *search and rescue operations* (predictions about potential search areas), and others.

Deployment of ML models in high-stakes scenarios increasingly requires explanation of ML models [Rudin, 2022; Freiesleben, Grote, 2023; Albahri, et al., 2023; Sahoh, Choksuriwong, 2023; Ghassemi, et al., 2021]. Explanation must be *exact* (accurately describe the model's inner workings and decision process) and *convincing* to the user [Rizzo, et al., 2023]. If one of these properties is lacking, then it is only a quasi-explanation not serving the explanation goal [Kovalerchuk, et al., 2021]. Several deficiencies of current interpretable machine learning (IML) methods are summarized in [Freiesleben Grote, 2023] from the literature: (1) LIME and SHAP, and counterfactual explanations can all be tricked to provide *any desired explanation* by modifying the model in data support. (2) Saliency-based techniques are highly *unstable* to constant or small *noise* in images, (3) feature importance *scores vary* across equally well performing models. Therefore, many current popular AI/ML explanation methods are only quasi-explanations. We present current methods with their benefits and deficiencies including ways to overcome deficiencies with a focus on high-stakes AI/ML tasks.

Explainable AI/ML is much more than allowing humans to *understand* why certain decisions are made, but also matching this understanding with users' *domain knowledge* to gain a *confidence/trust* in the model. This requires a **human-in-the-loop**, which includes checking *consistency* of the ML model and the domain knowledge. Finally, explainable AI/ML can lead to *reliable* and *trustworthy* models. The success a human-in-the-loop heavily depends on human cognitive abilities and limitations. This makes **human visual knowledge discovery** with **granular computing** and **visual granular patterns** critical for the progress of interpretable machine learning. Therefore, a

significant part of the paper is devoted to discovering interpretable ML models with human visual knowledge discovery methods.

Unfortunately, some black-box models have been deployed without proper explanation increasing risk of wrong decisions. The negative consequences for high-stakes decisions include financial losses, harm to the patient and legal liability for the healthcare provider, traffic accidents and injuries, wrongful arrests, and unjust punishment.

Major research topics in interpretable AI/ML models for high-stakes tasks include but are not limited to the following topics:

1. Providing *explanation and interpretation* of *learning models,* including deep models.
2. Development of *interpretable feature selection* and extraction techniques.
3. Efficient use of transparent and explainable methods like *decision trees* and *rules*.
4. Development of transparent and interpretable *ensemble methods*.
5. Exploiting *domain knowledge* to improve the interpretability and explainability of ML.
6. Investigation of the *trade-offs* between model complexity and interpretability.
7. Development of *visualization* techniques for explaining the decision-making processes of ML models.
8. Development of *visual knowledge discovery* processes for discovering ML models visually in *lossless visualization spaces.*
9. Application of explainable AI techniques to *real-world high-stake problems*, such as medical diagnosis, financial decision-making, and criminal justice.
10. Investigation of the impact of model interpretability on *user trust* and *acceptance* of AI systems.
11. Development of probabilistic and quantitative explanations for models with *uncertainty* and *risk* assessment.

Often interpretable models include decision trees, rule-based systems, and then these are combined with natural language processing. These models have been designed as interpretable originally. In contrast, models that are not interpretable originally (black boxes) need interpretation. This is very challenging for many high-stakes tasks.

The topics of emerging importance in interpretable AI/ML models for high-stake tasks include:
1. *Methodology* of explaining black-box models: It is often difficult to understand how these models make decisions, which is critical for high-stake tasks.
2. *Computational methods* to explain AI/ML models: Multiple existing methods can help to identify which features of a model are most important for making a particular decision, and unfortunately identified features can mislead in this. This is applicable to popular methods like LIME and SHAP.
3. *Human-in-the-loop*: This is a mandatory part of explainable AI/ML that is heavily underdeveloped. It can help to ensure that decisions made by the model are explained and understood by humans and will be trusted.
4. *Visual Knowledge Discovery* (**VKD**) is emerging as a major approach for implementing human-in-the-loop, where the human expert can provide valuable insights and feedback in the visualization space to build better models and prevent catastrophic errors.

Visual Knowledge Discovery is a process of visualizing and exploring complex high-dimensional information [Kovalerchuk, 2018]. It involves using visual cues, characteristics and properties to uncover hidden patterns and information. Major Topics in VKD are design and development methods to:

- *Identify patterns*, connections, and relationships between data points in the high-dimensional lossless visualization spaces with a variety of image recognition, machine learning, and visualization algorithms.
- *Represent and explore data* with feature extraction, clustering, dimension reduction and visual cues.
- *Conduct visual search* to help users quickly find relevant information in large datasets with image search engines, clustering algorithms, and visualization libraries.
- *Discover tacit useful knowledge* from large datasets using visualization techniques by user's interaction with high-dimensional data to find patterns, trends, and relationships in visualization spaces.
- Provide *visual query* with graphical user interfaces to interact with data and retrieve information using visuals, rather than textual queries.

It is shown in [Lakkaraju, Bastani, 2020] that explanations of black boxes can mislead users. They concluded that user trust can be manipulated by high-fidelity, misleading explanations stating that adversarial actors can fool end users into trusting an untrustworthy black box that uses prohibited attributes. To solve this problem these authors advocate for: (1) explanations as an **interactive dialogue** where end users who can query or explore different explanations (called perspectives) of the black box, and (2) capturing **causal relationships** between input features and black-box predictions], because, relying on correlations can be misleading and unrobust. Our V**isual Knowledge Discovery**

approach [Kovalerchuk, 2018] is fully within this paradigm. We advocate for interaction with lossless data and ML model visualization, as a natural and efficient way of interactive dialogue.

Development of new ML systems for safety-critical applications is urgently needed that are robust to the **most unfavorable conditions** to ensure that they continue to work *correctly* in a worst-case scenario [Lin, 2023; Robey et al.,2022; Huang et al., 2022]. **Correctness** expressed by the accuracy of ML models under worst-case scenarios on the available validation data is only one of the requirements. We also need to avoid data **overfitting** and data **underfitting** (**overgeneralization**) [Theisen, 2023], which are especially challenging under distributional shift, low-data availability, and imbalanced data distribution. It requires considering these issues in conjunction, analyzing how much each available method addresses others of these issues, and how to integrate these methods to get their synergetic effect. Note that **most unfavorable extreme conditions** like adversarial attacks or unexpected inputs that form worst-case scenarios do not represent the typical data. Thus, we need all three accuracy estimates: worst, average, and best-case estimates to get a full picture as formally presented in section 3.4.

The current methodologies for ensuring the accuracy of ML models under worst-case scenarios explore (1) **adversarial** robustness to **small changes** to input data, e.g., [Yang, et al., 2021] and (2) **minimax optimization**. The latter includes our approach with **Shannon function** as mathematically described in section 3.4 and **Worst-Case Feature Risk Minimization** (WFRM) [Lei, et al., 2023]. The WFRM approach is applied at each training iteration in the feature space helping to **resist overfitting** under distributional shift and low-data availability. The need to consider worst-case accuracy in addition to average task accuracy was also recognized for multitask learning [Michel, et al., 2021] with development of the Lookahead-Distributionally Robust Optimization (L-DRO) method to improve worst-case performance in multitask learning. Visualization of worst-case inputs, such as adversarial examples or data points that are difficult for the model to classify accurately, helps to understand the model and guide its improvement.

1.2. Historical aspects: prior 50 years of AI

Today, progress of artificial intelligence is mostly associated with deep neural networks (DNN). This is in sharp contrast with the logical reasoning core of AI for the past 50 years, dating back to 1956 when John McCarthy coined the term artificial intelligence. The Stanford Springe AI Symposium in 2006 was a sober celebration of 50 years of AI. One of the sessions of this symposium, which attracted a lot of attention, had a very illuminating title, "What went wrong and why: lessons from AI research and Applications." First, John McCarthy was asked to address this question. Below is a recollection of the author of this paper who attended that meeting. In short, McCarthy answered that AI's promise did not materialize because 50 years is too short for this to occur, and many more years are needed. He also listed some bad ideas that have been tried but which did not produce the expected results, among them he listed neural networks. Note this was told just a few years before current dramatic progress of DNNs started to materialize.

One can say that McCarthy was right, that more time is needed, pointing out that the current progress based on DNNs is mostly for tasks, which have a relatively *low cost of individual errors* like conversion from speech-to-text, or recognizing animals or mountains in pictures for entertainment. The applications are scarce and mostly at the experimental stage for high-stakes tasks, with a heavy price for individual errors like cancer diagnostics. Soon after that symposium, the author asked Lotfi Zadeh to comment on McCarthy's statement of how 50 years is too short a time for AI development. He answered that the issue is not just a short time in itself, but a classical logic approach promoted by McCarthy, stating that instead a fuzzy logic approach would be more appropriate. Note that Zadeh started fuzzy logic in 1965 with coining this term. This field reached its 50 years in 2015. Its competition with DNNs is not obvious.

Discussions on both AI based on the classical logic and fuzzy logic lead us to a new question: "**What is failing now in AI based on DNN models?"** It seems that the answer is that **explanation of black-box models** for high-stakes tasks is missing. Explanation was a core goal of the prior 50 years of AI both based on the classical logic advocated by McCarthy and fuzzy logic advocated by Zadeh. How did it happen that this focus was sidelined by DNNs which produce complex black-box models? For a long time, the **accuracy** of the models was a focus at many Neural Network (NN) conferences, with sidelined interpretability issues, because inaccurate models would be quite useless. Then, when the accuracy of many models reached a desired level DARPA started the eXplainable AI (XAI) program in 2016 [DARPA, 2016] attracting attention to the interpretability issue. Another motivation of XAI is that the high accuracy was reached with multiple caveats. An illuminating example is a panda picture recognized by the model as a gibbon monkey picture [Goodfellow et al, 2014] after adding a small amount of noise to the image, which was imperceptible for the human eye. We should not equate uninterpretable methods with neural networks. Many other methods in machine learning, including classical linear models, are of questionable interpretability, which we will discuss later for heterogeneous data.

One can say that the XAI program is a return to the original intent of AI as started by McCarthy. The following is an extract from this DARPA program: "The goal of Explainable Artificial Intelligence (XAI) is to create a suite of new or modified machine learning techniques that **produce explainable models**…In the early days of AI, the predominant reasoning methods were logical and symbolic. …Yet these early systems were much less effective… Success came as researchers developed new techniques. …These new techniques include support vector machines, random forests, probabilistic graphical models, reinforcement learning, and deep learning neural networks. Although, these more opaque models are more effective, they are less explainable."

While the declared goal of XAI is techniques which **produce explainable models,** the real focus as for now (2024) is attempting to explain black-box models, which is very different from producing new **accurate and explainable models without black-box producing techniques**. In fact, it's difficult to list such new explainable models that are fundamentally different from those developed in classical AI. The problem with the current dominant approach to explaining black-box models is that it is not clear if it can be reached beyond quasi-explanations [Kovalerchuk, et al., 2021] for difficult high-stakes tasks, situations where users must appropriately trust models. The development of an effective explanation interface was also one of the goals of DARPA XAI program with "the appropriate use of **visualization** and natural language understanding", and "dialogue to allow the user to interact with and drill down into the explanation." It was pointed out that, "breakthroughs are more likely to come from the right combination of machine learning and Human-Computer Interaction (HCI)" [DARPA, 2016]. The emerging explainable Visual Knowledge Discovery studies [Kovalerchuk, 2018; Kovalerchuk, et al., 2022] analyzed here are within this paradigm.

1.3. Scope

The scope of this paper is defined by the fact that complex black-box Machine learning (ML) methods are increasingly used for making high-stakes decisions. The end-user must trust ML models and assumptions used to produce ML models. We are far away from the trust required. We consider two open problems respectively: **P1: Explanation of ML models in High-Stakes Scenarios** and **P2: Reliability of ML models in High-Stakes Scenarios**.
Table 1 contrasts high-stakes with low-stakes ML tasks. It shows much higher trust requirements for high-stakes tasks relative to the low-stakes tasks. A typical example of high-stakes tasks is cancer diagnostics and for low-stakes tasks a typical example is optical character recognition (OCR) of the text printed from a laser printer.

Table 1. Comparison of typical high-stakes and low-stakes tasks.

#	Characteristic	High-stakes task	Low-stakes task
1	Cost of individual error	Higher	Lower
2	Available training data	Smaller	Larger
3	Imbalance of classes in training data	Higher	Lower
4	Model overgeneralization	Not acceptable	Acceptable
5	Model overfitting	Not acceptable	Acceptable
6	Model accuracy by average number of errors	Not acceptable	Acceptable
7	Model accuracy by worst number of errors	Required	Not required
8	Explanation of model	Required	Not required
9	Low quality of model explanation methods	Not acceptable	Acceptable
10	Current quality of model explanation methods	Insufficient	Sufficient
11	Extensive domain expert knowledge to accept model	Required	Not required
12	Use of model	Second opinion	Primary decision
13	Current level of model use	Experimental	Industrial scale

Several studies have been conducted in this area. The goal of this study is to **integrate** them together based on the combination of Visual Knowledge Discovery and worst-case estimates for reliable and interpretable machine learning. This paper is organized as follows. Section 2 is devoted to problem P1 (explanation of ML models in high-stakes scenarios). In sections 2.1 and 2.2 we show that many explanations are rather quasi-explanations than real explanations. Then, we present interpretable linear models beyond quasi-explanations (section 2 3), problems of linear models in deep learning (section 2.4), and explanations beyond quasi-explanations (section 2.5).

Section 3 is devoted to problem P2 (reliability of ML models in high-stakes scenarios). In section 3.1 we show that often the common evaluation of accuracy ML models is exaggerated leading to data overgeneralization, which is unacceptable in applications with high cost of individual errors. Learning with Reject Option (LRO) to counter overgeneralization is presented in section 3.2. The methods that avoid unreliable predictions with worst worst-case accuracy estimates are presented in section 3.3 and mathematical formulation of the worst-case estimates with Shannon functions are presented in section 3.4. Section 4 summarizes the paper with broad conclusion.

2 Explanation of ML models in High-stakes Scenarios

2.1. Overview and quasi-explanation vs. real explanations

The **typical** outputs of **explanations** of AI/ML model are:

1. The order, or rank, of *feature importance/salience* for prediction of individual cases or their associated sets.
2. A model in the *understandable* form (decision tree or logical rules).
3. A *simplified* model like a linear model (as a form of "shortcut").
4. *Similar cases* for the case to be predicted (factual explanation).
5. *Dissimilar cases* from the opposite class highlighting the difference of cases (counterfactual explanation).

Unfortunately, many on these explanations are **quasi-explanations** rather than real ones as we will show below. For model explanations to be real they should be **understandable for domain experts,** with model's decision process made clear for them to have a chance to **trust** the model outputs. To have an understandable model is a **necessary,** but **not a sufficient** requirement, for the model to be both **trusted** and **deployed** by the end user.

The desired properties of explainable AI machine learning models for humans are well documented in the literature for a long time [Craik, 1967; Doshi-Velez, 2017; Kovalerchuk, et al., 2021]. The explanation should be **comprehensible** to humans in *natural language* and *easy to understand representations*. The explanation should be within **domain knowledge** making *sense to a domain expert* without using terms that are *foreign* for the domain like distances, neural network layers, activation, and so on. Figure 1 summarizes the key current topics in explainable AI/ML tasks, which we have divided into three categories: methodology, specific methods, and applications.

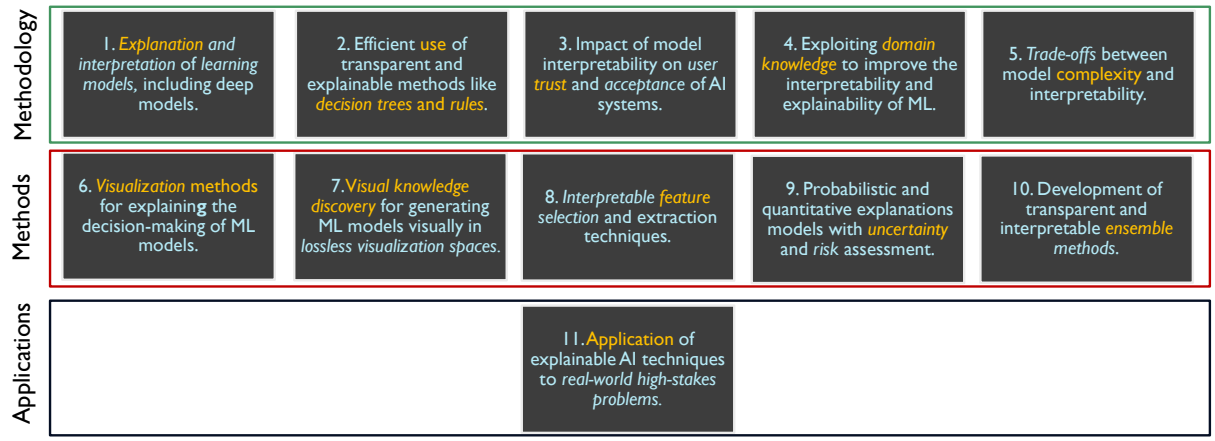


Figure 1. Topics in explainable AI/ML tasks.

The first topic in Figure 1 is explanation and interpretation of learning models. It is represented by two fundamentally different approaches:

Approach 1: Build accurate explainable model in the first place such as self-explained decision tree and logical rules.

Approach 2: Explain accurate shallow and deep black-box model like Support Vector Machine (SVM), Random Forest (RF), some linear models, and Convolutional Neural Networks (CNN).

Figure 2 illustrates these approaches.

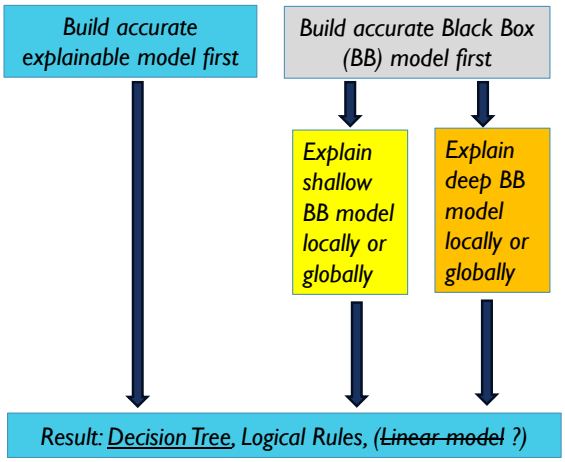


Figure 2. Explanation approached

Above some **linear predictive models** are listed in the category of the black-box model, while in many publications linear models are considered as *very interpretable* models due to their *simplicity* with considering their coefficients as *feature importance* [e.g., Molnar, 2020]. While it makes sense for **homogeneous features** which have the same physical meaning as temperature at different time moments. However, in machine learning, typically data are **heterogeneous** with one feature measuring temperature and others measuring blood pressure, pulse, and so on. This explains crossed "linear model" with a question mark for it in Figure 2 to attract attention to this important issue.

We have two categories of linear models: **linear predictive models** and **linear explanation models**. *Linear predictive models* are often considered as interpretable by default, which do not require any further explanation. Respectively, this assumption that linear models are perfectly explainable is a basis of *linear explanation models* in many popular methods like LIME, SHAP, and their different modifications. We show below that it is an extreme simplification of the actual situation with linear predictive models.

Can we trust a model that adds 3 kilograms and 5 meters in physics or 3 molecules and 30°C temperature in chemistry? These sums have no meaning and explanation in the natural sciences for such heterogeneous data. Respectively, weighted sums like 0.2*3kg + 0.6*5m and 0.2*3 molecules + 0.6*30°C have no meaning in the natural sciences for the heterogeneous data too.

In contrast, ML linear models freely use weighted sums like $0.2x_1 + 0.6x_2$ of heterogeneous data in linear regression, linear discrimination functions, Support Vector Machines, shallow and deep Neural Networks, Principal Component Analysis, and in ML explainability methods like LIME and SHAP. The **abstract mathematical symbols** x_1 and x_2 **hide** that the data are heterogeneous. What is the meaning of such sums in ML? The questionable explainability of many complex ML black-box models is rooted in this mathematical hiding of data heterogeneity.

The resulting explanation will only be a **quasi-explanation** [Kovalerchuk, et al., 2021] therefore, not a real explanation of linear models with such heterogeneous data. Other deficiencies to interpretability of linear models are also documented in the literature. One of them is the **sensitivity of coefficients** when attributes are very correlated. This results in very different importance attributes in linear models built on the same data. There are even examples where the coefficient of the same attribute is positive in one linear model and is negative in another linear model built on the same data. Already quoted deficiencies of current interpretable machine learning (IML) methods are that they (1) can be tricked to provide *any desired explanation,* (2) can be highly *unstable* to noise, and can *vary* across equally well performing models [Freiesleben, Grote, 2023].

2.2. Linear quasi-explanations: SHAP Tree-Explainer for tree based models

The polynomial time algorithm SHAP Tree-Explainer for tree-based models to compute explanations as feature importance values based on the Shapley game theory assumptions was proposed in [Lundberg, et al., 2019]. It is intended to measure a local feature impact and interactions. It is stated in [Lundberg, 2019] that Tree-Explainer

matches human intuition across a benchmark of 12 user study scenarios. However, it is not free from important deficiencies making it rather a quasi-explanation method. Figure 4 shows a simple visualization with local explanations based on TreeExplainer of a non-linear combination set of gradient boosted decision trees [Lundberg, et al., 2019].

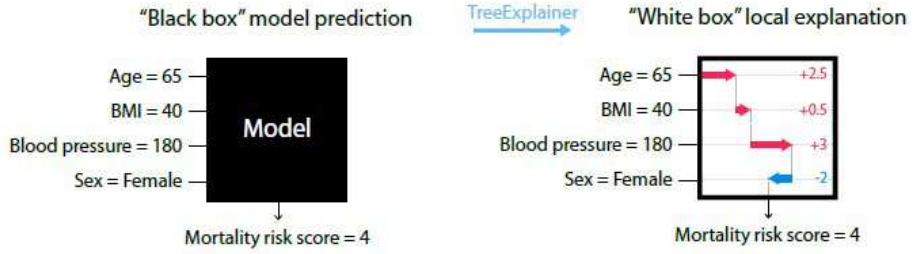


Figure 4. SHAP Tree-Explainer for tree-based models [Lundberg, et al., 2019]: linear explantion of mortality risk score 4 = 2.5*(age) + 0.5*(BMI) + 3*(blood pressure) – 2*(sex).

This method was designed for tree-based models, which includes the standard decision trees and random forests. While for random forest we really need explanations but for the regular decision tree it is not needed because they already have clear explanations. Moreover, explaining a nonlinear decision tree with linear functions can oversimplify and distort the impact of nonlinear dependencies between attributes.

Figure 4 shows a Tree-Explainer example of a black-box tree-based model for a given patient. It explains the Mortality Rate Score, MRS(p) = 4, f for a female patient *p* of age = 65 with BMI = 40 and blood pressure = 180. Here in the explanation MRS(p) = 4 is deciphered as a weighted sum 2.5*(age) + 0.5*(BMI) + 3*(blood pressure)-2*(sex) = 4. Is it a real explanation or a quasi-explanation? First it assumes that contributions of **heterogeneous attributes can be summed** up. This assumption came directly from the Shapley game theory assumptions in **economics** not from the medical domain. In economics the contributions are typically homogeneous, e.g., dollars, which is not the case in this medical example. This is a first indication that this example is likely only a quasi-explanation.

Next, obesity (BMI = 40) could cause the high blood pressure = 180 known as obesity-induced hypertension. However, in this example the impact of BMI is only 0.5 vs. 3 for the blood pressure. Thus, BMI is 6 times less important than the blood pressure. While the immediate cause of death can be hypertension, obesity can be its actual cause, which is not captured by this explanation, missing a deep causal effect.

Another source of a weak quasi-explanation in this example is mortality risks score (MRS) itself. This is assigned to an individual patient, but it is not physically measured like BMI or blood pressure for this patient. Creating an extra uncertainty to interpret it as a linear heterogeneous weighted sum. In general, the use of score concepts like MRS in the medical domain is caused by the lack of better measured characteristics.

Unfortunately, these important negative linear aspects of SHAP and Tree-Explainer are not highlighted in the literature, while the base SHAP paper [Lundberg, Lee, 2017] was cited over 18,700 times by January 2024. The literature emphasizes its positive aspects like consistency/monotonicity and computational benefits of TreeExplainer [Lundberg, et. al, 2020].

The benefit of TreeExplainer is in providing a compact summary of some properties of the tree-based models by showing attributes. Presumably shown attributes contribute most significantly to the success of prediction by multiple trees, but the concept of significancy is borrowed from economics with homogeneous attributes, which is not the case in the medical example shown above.

Below we list typical deficiencies of popular model explainers like SHAP and LIME linear explanation models. From the view of scientific rigor, the following shortcomings limit the trustworthiness of algorithmic explanations [Watson, 2022]:

- Fail to meet minimal severity test to affirm the explanation and to reject its negation [Mayo, Spanos, 2006].
- Do not test the causal effects they infer.
- Do not evaluate the uncertainty of their outputs.
- Do not inform about bounds of their region of relevance.

In [Watson, 2022] these deficiencies are illustrated for local explanations, where popular additive explanation methods LIME and SHAP exemplify these deficiencies with

- A very small linear explanation region when the decision boundary is extremely nonlinear.

- Highly unstable coefficients that can get positive, negative, or zero values depending on region size.

Shapley values in SHAP are not a natural solution to the human-centric goals of explainability [Kumar, et al., 2020]. Expected relative ease of explanation by linear models like LIME and SHAP is not justified for heterogeneous data [Kovalerchuk, et al., 2021]. The explainer output is augmented with bounds and accuracy/fit computations using resampling [Fryer Strümke, 2021], but not solving the key issue of model misspecification as a linear one. A deeper alternative is capturing target function's topology near the input point and adding statistical tests to evaluate relationships [Fryer Strümke, 2021].

2.3. Interpretable linear models

Below we show how interpretable ML linear models can be built. We define the criteria when the linear model can be justified, and when it cannot be justified, therefore requiring modifiation. Consider an example of a simple linear classifier,

$$f(\mathbf{x}) = ax_1 + bx_2 = 2x_1 + 2.2x_2,$$

$$\text{If } f(\mathbf{x}) > 12, \text{ then } \mathbf{x} \text{ belongs to class 1, else } \mathbf{x} \text{ belongs to class 2,} \quad (1)$$

where $\mathbf{x} = (x_1, x_2)$. Let $\mathbf{u} = (2, 5)$ and $\mathbf{w} = (5, 2)$ be two 2-D points with values $f(\mathbf{u}) = 2*2 + 2.2*5 = 15$ and $f(\mathbf{w}) = 2*5 + 2.2*2 = 14.4$ where both values 15 and 14.4 are greater than a threshold of $T = 12$, respectively 2-D points **u** and **w** are classified to class 1. The presence of **u** and **w** in the training data demonstrates that attributes x_1 and x_2 are **mutually exchangeable**. A smaller value of $x_2 = 2$ in **w** relative to $x_2 = 5$ in **u** is **compensated** by increasing the value of $x_1 = 5$ in **w** relative to $x_1 = 2$ in **u**. This compensation is a key property of linear or non-linear polynomial discrimination models, leading to a solution of the **model justification task** for linear and polynomial classifiers. The attributes x_1 and x_2 can be **heterogeneous** like blood pressure and BMI, but the presence of points **u** and **w** tells us that despite their different physical modalities, their impacts on the target attribute are compensatory as an empirical fact.

The **dual task** is finding coefficients *a* and *b* of function *f*. It differs from the standard machine learning task of finding linear discriminant function (LDF) which does not check for the presence of points **u** and **w** with the compensation effect.

When training data contain points **u,** but do not contain compensatory points **w**, we should not build LDFs in a standard way and deliver it to the domain expert. Such LDFs likely will **overgeneralize** training data to **w** points. This is a likely reason why a domain expert has **difficulties** to understand and interpret linear model in this situation, where the experts' domain practice does not contain compensatory cases **w** for cases **u**.

This analysis guides us on developing an algorithm to build a **Justifiable Linear Model** (JLM algorithm) with the following main steps:

Step 1: For each case like **u** = (2.5) *search* for similar cases, like **z** = (2.3, 4.9) and for the compensatory cases like **w** = (5, 2).

Step 2: If the number of **w** cases is *comparable* with the number of **z** cases, then we build a standard LDF on these data.

Step 3: If the number of **w** case is *much smaller* than the number of **z** cases, then we build **another model constrained** LDF (**CLDF)** on these data as explained below.

Step 4: *Convert* models built on Steps 2 and 3 to alternative interpretable models without weighed summation of attributes, like decision trees, logical rules and hyper-rectangles (hyperblocks).

This model is restricted by additional requirements on intervals of attribute values of cases to control model overgeneralization to **w** cases. We provide these constraints by further developing our hyperblock approach [Huber, et al., 2024, Hayes, et al., 2024], where before we built hyperblocks from LDF without excluding **w** cases.

The idea of the constrained LDF algorithm is as follows. To exclude the case **w**, we limit values of attributes x_1 and x_2, by vicinities of their values for case **u**, e.g., $x_1 \in [1, 3]$ and $x_2 \in [4, 6]$. These intervals include **u** = (2, 5) but exclude **w** = (5, 2). Respectively, the CLDF model will be

$$\text{If } f(\mathbf{x}) > 12 \ \& \ x_1 \in [1,3] \ \& \ x_2 \in [4,6], \text{ then } \mathbf{x} \text{ belongs to class 1.} \quad (2)$$

Here the cases which do not belong to class 1 do not automatically belong to class 2. For class 2 we need to add similar constraints. Then we can build a full classifier, which will likely have three categories: class 1, class 2, and refuse (cases, which do not belong to classes 1 and 2). The property $x_1 \in [1, 3]$ & $x_2 \in [4, 6]$ is a square/rectangle in 2-D space, which is generalized to a **hyperblock** HB in n-D space with each attribute x_i in its' limits L_{i1}, L_{i2}, $x_i \in [L_{i1}, L_{i2}]$.

Formula (2) covers cases **u** and **z** and excludes **w**, but training data contain many more cases. The CNDF can use an iterative approach, by finding training cases that are not in the current HB, then adding those HBs to (2). This creates a **constrained LDF (**CLDF) in a general form:

$$\text{If } f(\mathbf{x}) > T \text{ \& } (\mathbf{x} \in HB_1 \text{ or } \mathbf{x} \in HB_2 \ldots \mathbf{x} \in HB_k) \text{ then } \mathbf{x} \text{ belongs to class 1.} \quad (3)$$

This classifier is applicable to n-D data. It is not a pure LDF anymore, but it is a constrained LDF. The algorithm proposed in [Huber, et al., 2024] discovers a set of hyperblocks from a given LDF $f(\mathbf{x})$ that satisfy the class property: If $f(\mathbf{x})$ = class 1, then **x** belongs to one of those k hyperblocks. That algorithm can be applied at Step (3) of the JLM algorithm above to get hyperblocks needed in (3). Similarly, we can find hyperblocks for class 2. Step 4 *converts* models built in Steps 2 and 3 to alternative interpretable models without weighed summation of attributes can be performed by the same algorithm that builds hyper-rectangles (hyperblocks) in step 3 from [Huber, et al., 2024].

2.4. Problems of linear models in deep learning

While this paper mostly presents tasks with shallow machine learning models, in deep learning neural network models, the problem of underfitting/overgeneralization is the same. At the last layer **deep learning neural network models** use a linear model, or several of them, and have the same overgeneralization effect [Hein, et al., 2019] as we have discussed for the shallow models. These authors pointed out that it is a **fundamental inherent problem of the neural network architecture**. The expected solution is in modification of this architecture. "It would be desirable to have network architectures which have provably the property that far away from the training data the confidence is uniform over the classes: the **network knows when it does not know**" [Hein, et al., 2019]. In that paper the problem is posed as arbitrarily high confidence predictions of ReLU networks far away from the training data, which cannot be fully avoided by some known methods. It might (1) *take too many samples* to enforce low confidence on the whole out-distribution, and (2) still produce high confidence predictions in a neighborhood of *noise* images.

In [Hein, et al., 2019] a process called *confidence enhancing data augmentation* (CEDA) is proposed to deal with this problem with several theorems proved and experiments conducted. Theorem 3.1 states that ReLU networks always attain high confidence predictions far away from the training data. Then the construction from this theorem is used to experiment with uniform random noise images x and searching for the smallest α such that the classifier attains 99.9% confidence on α. They observed that the upscaling factor α is much higher for CEDA than for the plain model, concluding that CEDA improves the network far away from the training data.

We do not object to such modifications of the neural network architecture, but it has a fundamental problem of defining, "**far away from the training data**", without arbitrary assumptions while, "far away", is both task and data specific. It is well illustrated by our Iris example in Figure 11 as a classical fuzzy logic problem of **defining a linguistic variable** with values like, "far away", "nearby", and, 'intermediate". Note that, here we have a multidimensional linguistic variable, not one-dimensional linguistic variable, for individual attributes.

In Figure 11 we can compute the average distance between, "nearby", cases and build an envelope around these points on that distance. See a black oval for the Versicolor class shown in Figure 11. It is much better than a huge overgeneralization produced by the linear model. However, we cannot fully justify this distance-based approach because it requires experimental data that flowers can be only that far from the given ones inside of the oval. It is rather a biological not a mathematical issue.

In this example we use the distances between nearby points. Those points are well visible in Figure 11, and we can easily judge which points we consider nearby. We cannot do this easily in a typical **multidimensional machine learning space** because we cannot see those cases. The numerical values of distances do not give us the exact concept of, "nearby", and all clustering approaches are quite subjective. Thus, in the high dimensional space it is a fundamentally, "blind", approach bringing some **mathematical assumptions** that are, "**foreign**", for the task domain and domain experts. This fundamentally limits the trust that a domain expert can get from the black-box models.

A common approach, which assigns **probabilities** to every point in the feature space (with probabilities close to zero denoting points far away) does not fundamentally solve the problem either. We cannot accurately evaluate a multidimensional distribution using limited training data. For instance, for the Iris dataset, the number of values on individual attributes range from 18 to 59 with 2 digits per value. It results in the 4-D space with 573,480 points. For comparison, all the Iris dataset contains just 150 4-D points, which is 0.026% of the total number of points in this 4-D space.

Therefore, we advocate visual methods with **lossless visualization of multidimensional data** where we can much more realistically see where the data generalization limits should be. Our examples in this paper for shallow machine learning models show that this is doable. The same is doable for the last linear layer of deep learning approaches. This seems a promising direction to control underfitting/overgeneralization.

2.5. Explanations beyond quasi-explanations based on decision trees

2.5.1. Decision trees as main explanation models for black boxes

The consideration above shows that decision trees and logical rules are machine learning methods that are most widely accepted as being interpretable without controversy as associated with the linear models, which we discussed above. Therefore, decision trees and logical rules play a **critical role** in interpretability studies. The reasons are as follows. High-stakes machine learning models must be accepted by the domain expert to be deployed. It fundamentally requires a domain expert being involved in the model building and evaluation. Therefore, a domain expert (human) in the loop is mandatory. The visuals and natural language processing (NLP) are efficient ways to have a domain expert in the loop. Respectively the second topic listed in Figure 1 is efficient use of transparent and explainable methods like decision trees (DTs) and rules. This has the following challenges:

(1) How to *build* transparent explainable DT and rules as explainable models in the *first place*?
(2) How to build transparent DT and rules as *explainable interpolators* for the black boxes?
(3) How to *explain* complex DT and rules?
(4) How to put a *domain expert* in the driver seat of the decision tree and rules development and analysis?
(5) How to support tracing and analysis of *individual cases* in the DT and rules?
(6) How to support analysis of *similar cases* to the case to be predicted by the DT and rules?
(7) How to find *counterfactual cases* (cases closest to the case to be predicted but from the opposite class)?
(8) How to *modify* the DT and rules by the domain expert to get desired accuracy and confidence?
(9) How to support analysis that both training and validation data originate from the *same distribution*?

The known approaches to address these challenges are computational and interactive with extensive use of visualizations. The challenge (1) has been extensively addressed in the ML literature for decades with multiple algorithms available and new developments continue. We refer readers to this extensive literature. For (2) the major issue is getting accurate enough global and local approximation of the complex black-box models, which includes defining local areas and generating more data in them using those black-box models in question. The reason to generate more data is that often local original data used to build a black-box model are insufficient to build a robust local decision tree. Several methods are available to generate extra local data for SHAP and others [Lundberg, et al., 2017, 2019, 2020].

For (3), one of the approaches to explain complex DT and rules is discovering their most important features. The SHAP Tree-Explainer discussed above is within this approach. For a single decision tree, we don't need special methods for explanation because the Information Gain and Gini index, which are often used to build DTs, are good indicators of feature importance along with the DT structure.

For (4)-(9), which includes putting a domain expert in the driver seat of the DT and rules development and analysis, we will present **interactive Visual Knowledge Discovery** methods summarized from [Kovalerchuk, Dunn, et al., 2024]. It includes methods to represent DTs and n-D data losslessly with Bended Attributes and Shifted Paired Coordinates.

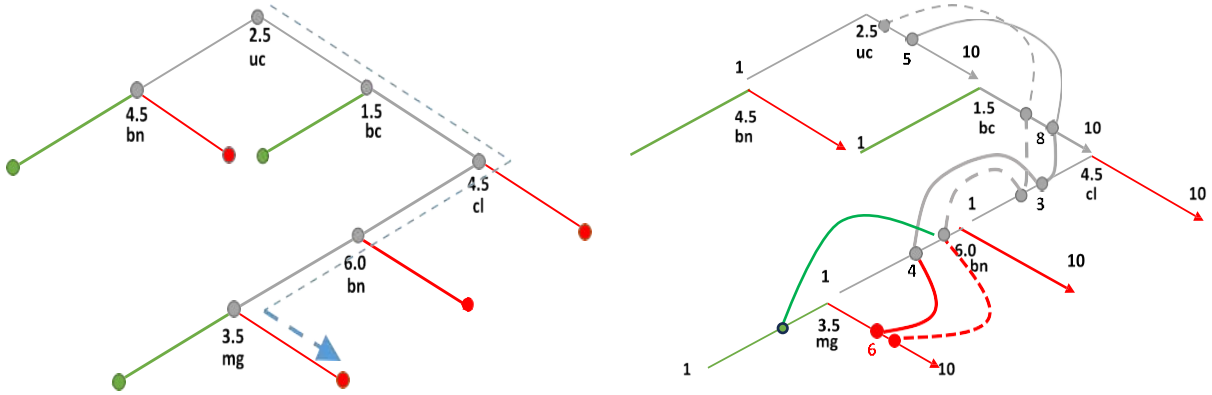
2.5.2. Bended Coordinates Decision Tree (BC-DT) method

Figure 5(a) illustrates a traditional visualization of a DT, where green edges and nodes indicate the benign class and red edges and nodes indicate the malignant class. It does not allow a domain expert to address (5) by efficient tracing and analyzing individual cases in the DT and rules. In Figure 5a, a dotted blue line traces the malignant cancer case, but it does not show actual values of factors and closeness of these values to the thresholds within each node.

In contrast Figure 5(b) shows a DT in a new **Bended Coordinates Decision Tree** (**BC-DT**) method [Kovalerchuk, 2023; Kovalerchuk, Dunn, et al., 2024], which shows to the domain expert both actual values of attributes, and how close these values are to the thresholds within each node. It is visible that the dotted case (patient) is closer to the threshold values in attributes uc, bc, and bn, and further in attributes cl and mg, which indicates the confidence/certainty in the DT decision relative to the distances to the respective attribute thresholds.

This visualization also helps to address (6), which is supporting analysis of similar cases to that of the case to be predicted by the DT and rules, when the dotted case is a new case that needs to be predicted, and a non-dotted case is available, then thecase is a similar case as from the training data. Also, this visualization addresses (7), which is supporting analysis of counterfactual cases (cases closest to the case to be predicted, but from the opposite class). The case shown with the green last curve belongs to the benign class and differs from the dotted case to be predicted only in this segment (mg attribute).

To address (8), which is supporting interactive modification of the DT and rules by the domain expert to get desired accuracy and confidence, the BC-DT method allows a user interactively *adjust thresholds*. To address (9), which is supporting analysis that training data and validation data are from the same distribution, the BC-DT method allows a user to visualize all training cases on one DT and all validation cases in another DT *side-by-side* for their comparison.

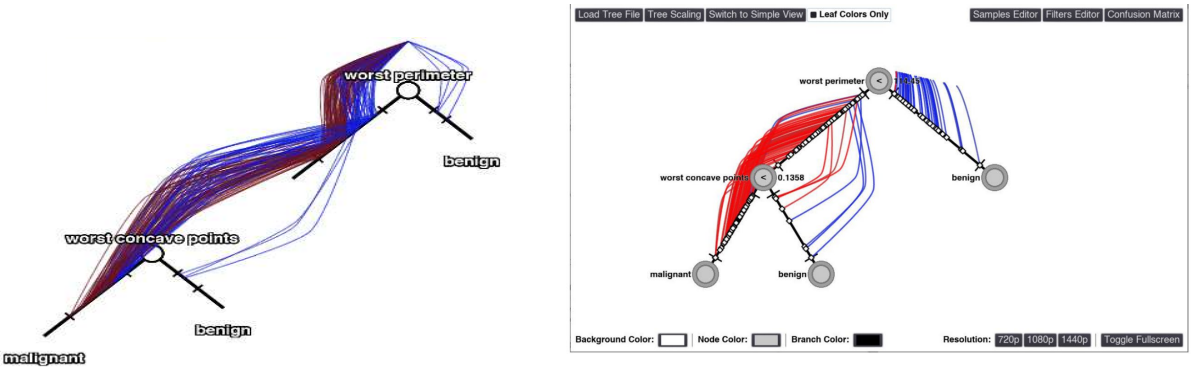


(a) Traditional visualization of a decision tree. Green edges and nodes indicate the benign class and red edges and nodes indicate the malignant class.

(b) DT with edges as Bended Coordinates in disproportional scales. The curved lines are cases that reach the DT malignant edge with different certainties due to the different distances from the threshold node.

Figure 5. Explanation with Bended Coordinate Decision Tree (BC-DT) [Kovalerchuk, 2023].

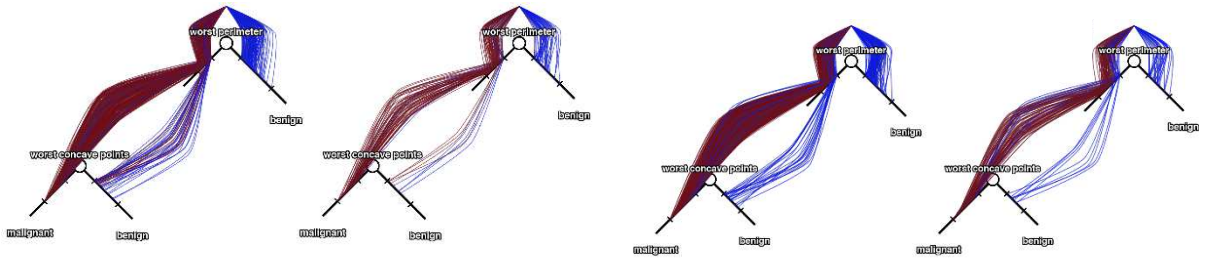
The idea of the Bended Coordinates Decision Tree (BC-DT) method is as follows [Kovalerchuk, 2023; Kovalerchuk, Dunn, et al., 2024]. In a DT the attribute that is associated with a node is bended at the threshold point as shows in Figure 5b, i.e., the edges of the DT graph serve not only as links between nodes but also represent respective coordinates as bended lines with the start point on the left end and the end point on the right end. This is indicated by the arrow. For example, the attribute uc at the root of the DT covers the interval [1, 10] with a threshold at 2.5. This means that the left interval is only [1, 2.5] and it is much shorter than the right interval (2.5, 10]. In Figure 5b these intervals are made equal by different (disproportional) scaling them to keep this visualization like a traditional visualization on Figure 5a. In the proportional scaling, the sides can be of different lengths if the threshold point is not in the middle of the interval. Figure 6 shows DT with cancer data for both options implemented.



(a) BC-DT with proportional (unequal) edges (b) BC-DT with disproportional (equal) edges

Figure 6. BC-DT with proportional edges (a) and disproportional edges (b) serving as Bended Coordinates [Kovalerchuk, Dunn, et al., 2024].

Figure 7 shows trained DT before and after eliminating false negatives for breast cancer data by interactive adjusting thresholds. It also shows how similar or different training and test data allowing a user to analyze their similarity.



(a) Trained DT (training data on the left) and test data on the right) (b) DT after eliminating false negatives (training data on the left and test data on the right).

Figure 7. Trained DT before and after eliminating false negatives for breast cancer data [Kovalerchuk, Dunn, et al., 2024].

2.5.3. Shifted Paired Coordinates – Decision Tree (SPC-DT) method

An alternative approach to put a human-in-the-loop with decision trees is illustrated below with **Shifted Paired Coordinates – Decision Tree** (**SPC-DT**) method [Worland, et al., 2022; Kovalerchuk, Dunn, et al., 2024]. The SPC-DT technique enhances and complements the capabilities of the traditional DT model visualization expanding them by visualizing: (1) **relations** between attributes, (2) **individual cases** relative to the DT, (3) **detailed data flow** in the DT, (4) the **tightness** of each split threshold in the DT nodes, and (5) 2-D **density** of cases in areas of the n-D space. Experts in the field can greatly benefit from this knowledge in the process of evaluating and enhancing DT models. To the best of our knowledge, none of the DT visualization techniques now in use provide all these capabilities.

SPC-DT is summarized in Figure 8. It uses the **Shifted Paired Coordinates** (**SPC**) representation, which is a category of **General Line Coordinates** (**GLC**) [Kovalerchuk, 2018]. SPC, as its name suggests, is a collection of shifted pairs of Cartesian coordinates. An n-D point **x** is represented in 2-D by a directed graph in SPC, where nodes are made up of consecutive pairs of **x**'s coordinate values (x_1, x_2), (x_3, x_4),…,(x_i, x_{i+1}),…, (x_{n-1}, x_n), and edges connect them. The example in the bottom of Figure 8 illustrates SPC. It shows the 6-D point $\mathbf{x} = (x_1, x_2,..., x_6) = (1, 2, 3, 2, 5, 1)$, where a first pair $(x_1, x_2) = (1, 2)$ is visualized in the first pair of coordinates, the second pair $(x_3, x_4) = (3, 2)$ is visualized in second pair of coordinates (X_3, X_4), and the third pair $(x_5, x_6) = (5, 1)$ is visualized in the third pair of coordinates. In Fig. 8, a directed graph (1, 2), (3, 2), (5, 1) is formed by connecting these three points. These pairs of coordinates are shifted relative to one another to prevent overlap.

To create a DT visualization from a given decision tree model, the SPC-DT process is as follows: (1) parsing the DT model, (2) pairing attributes, (3) building a set of paired Cartesian coordinates, (4) drawing each pair of coordinates in the default location shifted one over another, (5) mapping a part of the decision tree to a respective pair of coordinates as rectangles based on the threshold values, (6) coloring rectangles according to classes, and (7) drawing a set of n-D points as directed graphs in the SPC.

In the middle of Figure 8 shows a selected line of the DT description (green classification rule). This line was parsed from the DT description shown on the right. Then this rule is visualized in the first pair of Cartesian coordinates on the top of Figure 8 as the green area. Similarly, the red area in this pair of coordinates represents the second rule for this DT: If ucellsize < 2.5 & bnuclei ≥ 4.5, then class = *malignant*. The gray area in this pair of the coordinates represents cases that the DT did not classify yet. The visualization process is continued for the remaining nodes of the DT with the rules: If ucellsize ≥ 2.5 & bchromatin < 1.5, then class = *benign* (see the green arrow from the gray area in the first plot to the green area in the middle plot).

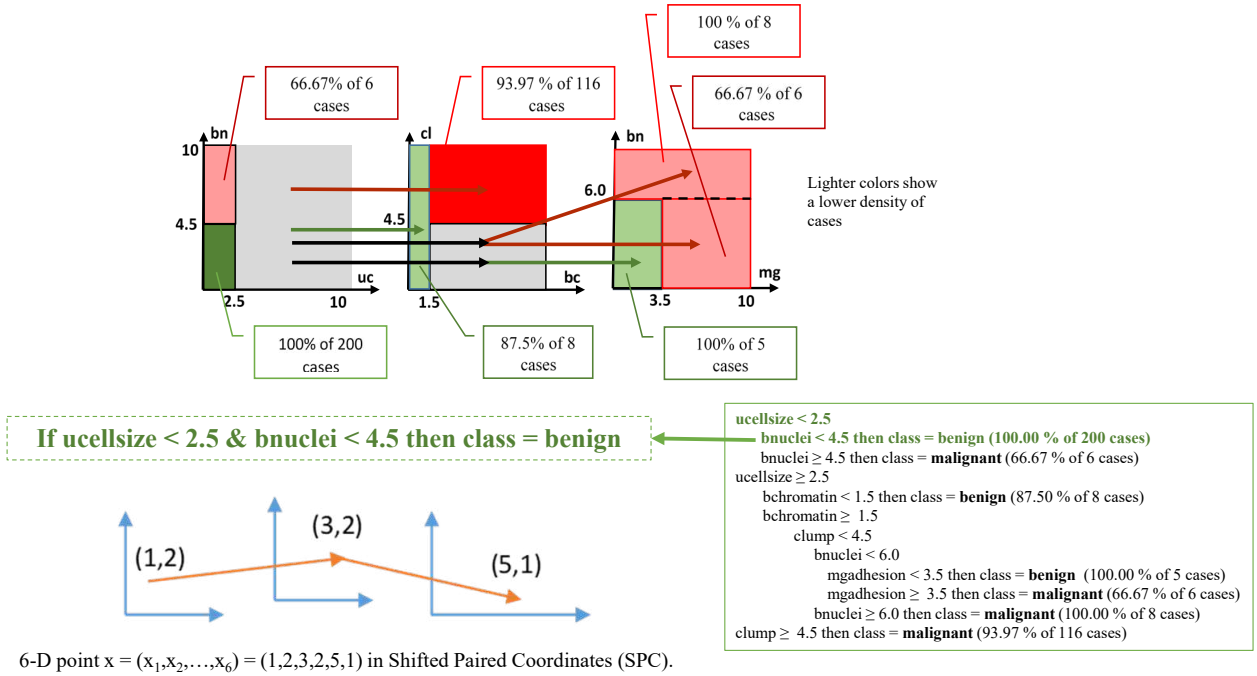


Figure 8. Shifted Paired Coordinates – Decision Tree (SPC-DT) method.

After a decision tree visualization is built according to this procedure a human-in-the-loop process can start with the following interactive capabilities: (a) change visualized n-D *points*, (b) *shift* the location of a coordinate pairs, (c) change class *colors*, (d) *inverse*/flip coordinates, (e) *swap* vertical and horizontal coordinates, (f) *condense*/shrink points in a rectangle, (g) show overall DT *structure*, (h) adjust *thresholds*, and (i) compute *accuracy* after adjusting thresholds. Figure 9 illustrates these capabilities for the cancer dataset to produce a user-friendly DT.

The gray areas on each pair of coordinates are areas where a case's class is unknown yet at the current DT level. To depict the various outcomes of cases in gray areas, several grayscale shades are used. The DT has classified some cases as, "malignant", while other cases have been classified as, "benign", as indicated by the red and green areas. A user can see the proximity of a case to each split threshold in the DT nodes. Moreover, graphs of training and test cases allow to see the density of their distribution in the n-D space. The number of cases in various areas of the space or the degree of *class purity* in the area is displayed using color intensity (lighter/darker).

The benefits of this SCP-DT visualization are in revealing (1) the **tree structure**, (2) **data flow** by arrows in the DT and (3) **relations** between: (3.1) *2 attributes* inside of each plot, (3.2) *4 attributes* using green and red arrows between two plots, (3.3) *all 6 attributes* using green and red arrows between three plots, and (3.4) *more attributes* for larger dimensions.

All these characteristics work together to boost trust in the DT model's ability to be understood by the end user and to predict accurately. Consider a DT where the actual training cases are all outside the threshold (border) area, but a new case that needs to be predicted is just on the border. Most likely, this DT model overgeneralized the training set, making the predictions excessively uncertain of such new case.

The interactive SPC-DT allows users to turn on and off, "context", attributes, which are attributes that are not part of the tree but are a part of the input dataset. The tree root can be connected to these attributes. When such properties are disabled, they can be grayed out or made semitransparent so that the context may still be seen. The SPC-DT design

allows showing just subsets of cases when we need to visualize many cases and classes. It can be, "centers", of subsets, min or max cases of subsets with the remaining cases being gray or semitransparent to decrease occlusion and simplify visual granular patterns. The SPC-DT design also allows showing the error rate for each dataset, branch of the tree, and the confusion matrix to aid the user in making confident predictions. Figure 10 shows a way to compare training and testing cancer data using two SPC-DT visualizations side-by-side. The distribution of testing cases differs from that of the training data. Significant areas covered by the training data are not covered by them. This can explain the diffence in accuracy. Here we have different accuracies: 91.43% on testing data and 95.39% on training data.

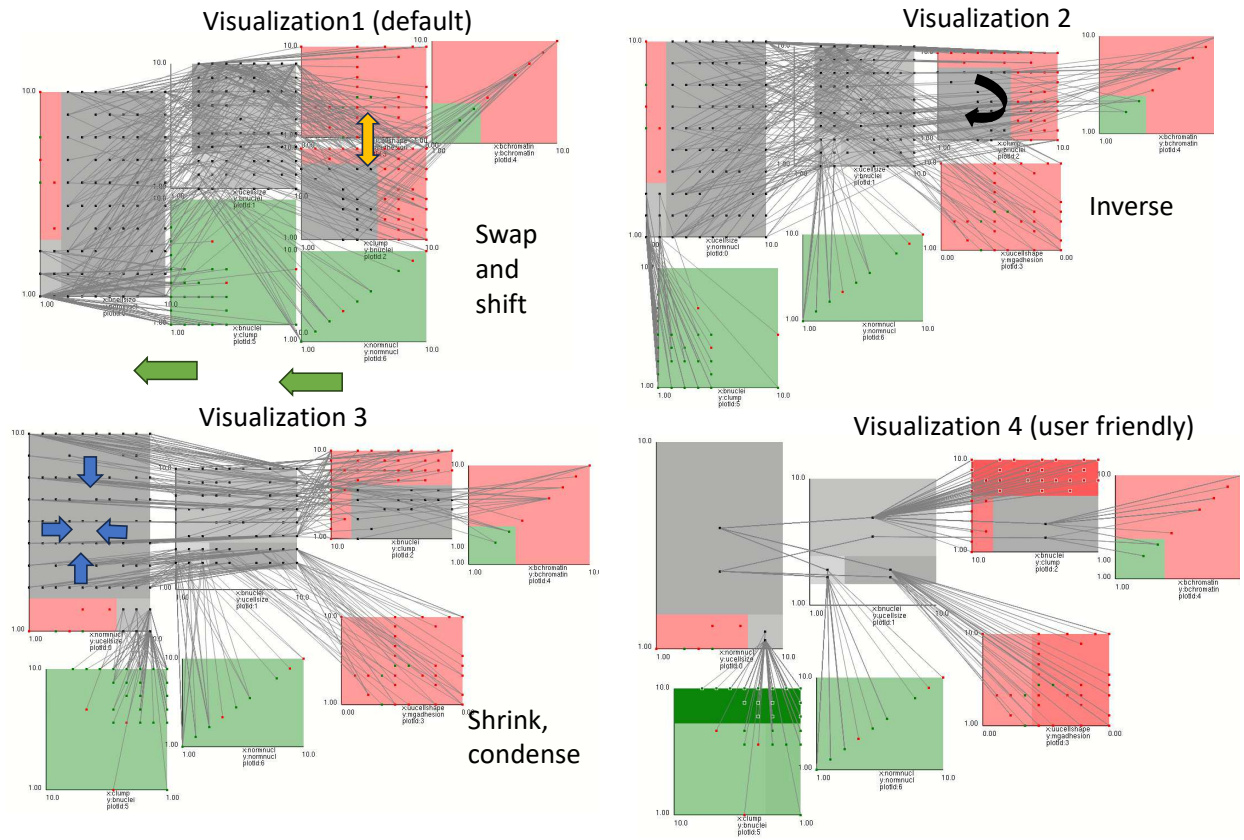
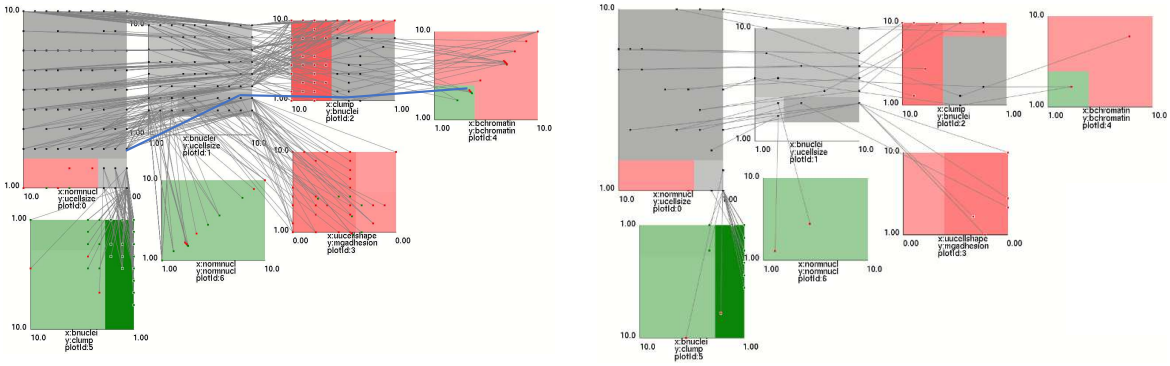


Figure 9. Illustration of interactive SPC-DT method for cancer data set to produce a user-friendly DT.



(a) Training data (90%) in SPC-DT (b) Testing data (10%) n SPC-DT.

Figure 10. Comparison of training and testing cancer data in SPC-DT with 90/10 split of all data to training and testing data.

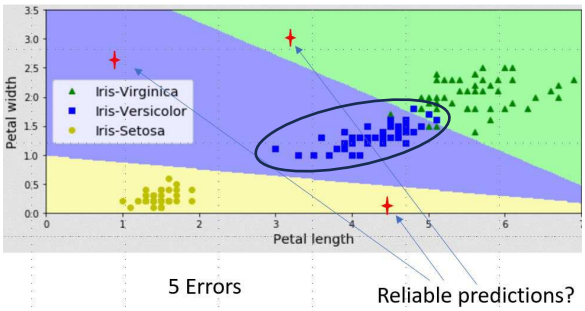
In summary BC-DT and SPC-DT visualization systems assist the user in understanding the DT and its performance in detail beyond a single accuracy value with support domain experts' ability to modify DTs if not satisfied. These

visualization systems aid domain experts in assessing the effectiveness, interpretation, overgeneralization, and overfitting of the DT model, which is critical for high-stakes prediction tasks.

3. Avoiding Unreliable Predictions for High- Stakes Tasks

3.1. Overgeneralization as a challenge for high-stakes task

Overgeneralization of data has multiple negative consequences for high-stakes tasks. This overgeneralization is one of the sources of unreliable prediction with exaggerated accuracy [Gao, et al., 2019]. Below we discuss these challenges and ways to overcome them. Figure 11a illustrates data overgeneralization by the linear Logistic Regression (LR) model. Red cases are far away from the actual cases of three Iris dataset classes. For instance, all real flowers of the Iris class Setosa have small length and small width of petal. There is no Setosa flowers with petal of medium length while LR classified such red case to this class. For the Versicolor class all cases are within the black oval, which is less than a quarter of the blue area of this class in Figure 11. While for the Iris dataset this error is likely not critical, but for high-stakes tasks it can be dangerous. For instance, the actual flood safe zone can be limited to a small area while the LR algorithm expands it far beyond. Adversarial activities is another source of danger of overgeneralization for high-stakes tasks.



(a) Logistic regression classification of Iris [Big data, 2018].

Setosa – small length and small width of petal
Versicolor – medium length and medium width of petal
Virginica – large length and medium to large width of petal

(b) visual and verbal Setosa, Versicolor and Virginica Iris petals.

Figure 11. Human visual and verbal Generalization vs. Machine Learning generalization: Iris dataset example.

It is a deficiency of LR and many other **ML algorithms** like Random Forest, Naïve Bayes, and Support Vector Machine (SVM) , which **classifies all points** in the feature space to one of the given classes, while some points of the feature space may not belong to any of them [Kovalerchuk Grishin, 2018; Kovalerchuk, 2023]. In the example LP allows long and short Versicolor Iris petals, which violate the "medium" petal length and width of Versicolor. Next, LR allows short *Virginica* Iris petal length that violates the "large" length of petal" of Virginica. It is very visible from Figure 11b, which shows examples of the actual iris petals and their verbal representations. Note that such overgeneralization cannot be resolved by adding more training data of given classes because cases outside the shown areas may not exist or may belong classes not presented in the training data. Figure 11a is not unique, many software ML packages like MLxtend [Raschka,, 2023] produce, "decision regions", without any warning how severe such, "decision regions", overgeneralize classes. A more detailed analysis of these issues can be found in [Kovalerchuk Grishin, 2018].

3.2. Learning with Reject Option (LRO) to counter overgeneralization

Below we present important approaches to avoid such undesirable overgeneralization. In general, the **constraint-based ML** paradigm is growing [Gori et al., 2023] as well as **learning with reject option** [Bartlett, Wegkamp, 2008; Zhang, et al., 2023]. Learning with reject option can be applied to high-stakes tasks [Gao, et al., 2019] if (1) it is backed up by a **human** or other default trusted methods and (2) performs better. Our approaches are within this paradigm as described below.

We will use the following terms. **Analytical ML models (AML)** are models built by *computational ML methods without using visual graphical tools.* **Visual ML models (VML)** are models built by using *visual graphical tools*. These models differ in many aspects, including **generalization** of ML training data, as we illustrated above with the

Iris dataset. Visual ML models allow to generalize more *conservatively* and *intuitively justified* more so than AML. **Visually inspired ML (VIML)** approach combines AML and VML. It *does not start with a predefined categories of functions* as linear or non-linear functions, which discriminate all points in the space to the given classes. It starts with *observation of the training data without any predefined function set*. VIML generalizes the training data more conservatively because of this observation [Kovalerchuk, 2023].

This leads to VIML **models**, which use the envelopes around the training data of each class, e.g., reflecting the small length and width of petal for Iris Setosa. Such algorithms **refuse** to classify points outside of the envelopes. How can we know that we need to use an envelope? We need tools to visualize *n*-D data in 2-D without loss of *n*-D information (**lossless visualization**) allowing to see *n*-D data as we see 2-D data. In the petal example, we have a linguistic term, "*small petal*", and two corresponding 2-D visuals: a picture of the small petal in Fig. 11b as a basis for the learning constrains and rejection.

These verbal and visual representations synergistically describe the same physical object in **intelligible** and **well-understood** ways, while both are quite uncertain and need formalization. Subjective probabilistic and fuzzy logic concepts are commonly used to formalize, "small", and to, "compute", and, "reason", with words. This synergy is going further and deeper to, "computing", with images [Kovalerchuk, 2013]. The Iris petal example illustrates an important property of the **explainable ML.** It can be in *uncertain, but understandable linguistic and visual terms*, not necessarily in the formal mathematical terms.

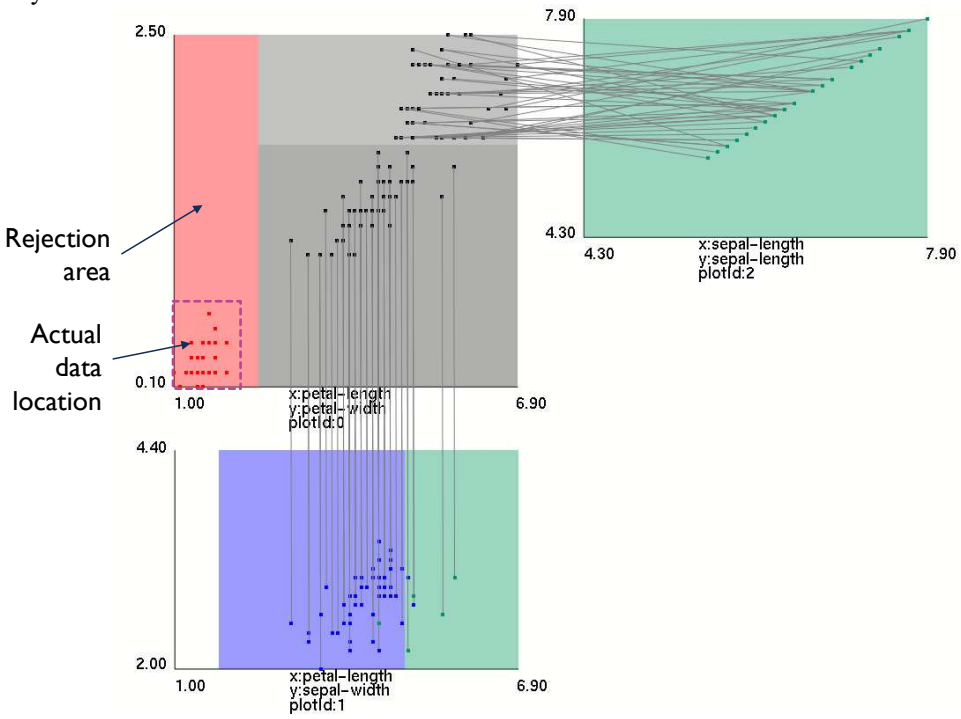


Figure 12. Learning with Reject Option (LRO) decision tree ML model in SPC-DT on Iris dataset.

Figure 12 illustrates the visual LRO approach. It uses the SPC-DT model visualization already described above and illustrated for cancer data in Figures 9 and 10. Now this is applied to the Iris dataset presented in Figure 11 with the difference from Figure 11 being Figure 12is showing more than only two attributes. Figure 12 captures these data in all four attributes losslessly in DT. The dotted rejection area within the red rectangle can be easily outlined by the user in this visualization. Similarly, a user can outline other rejection areas in Figure 12 to avoid overgeneralization. This illustrates the fundamental requirement for the visual approach that **n-D data are fully and losslessly visualized because** lossy dimension reduction methods can corrupt the n-D data leading to incorrect rejection areas.

It is important that outlining the **rejection and acceptance areas** is done by a domain expert who understands the meaning of the original attributes. It is much more difficult in artificial attributes produced by lossy dimension

reduction (DR) methods like principal component analysis (PCA), which are less interpretable or not interpretable at all for the domain expert [Kovalerchuk, Fegley, 2023].

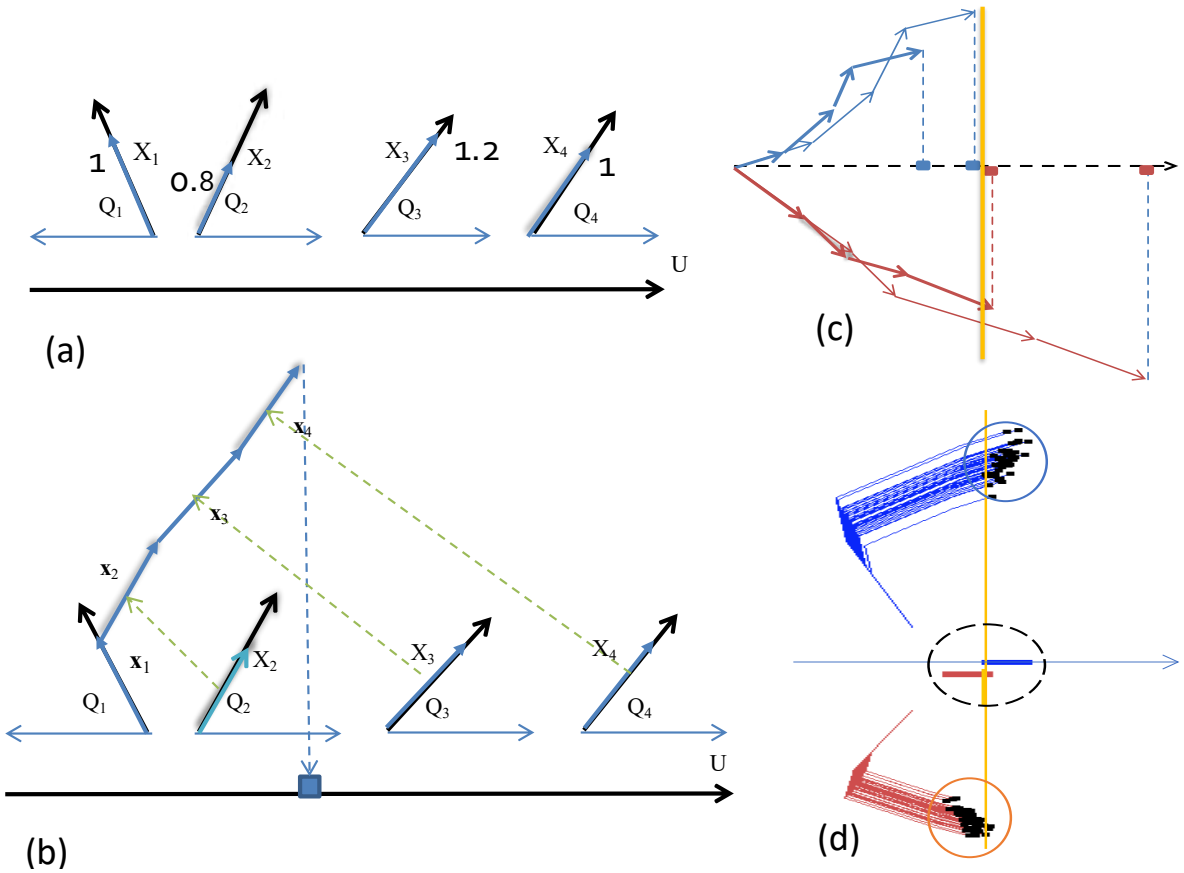


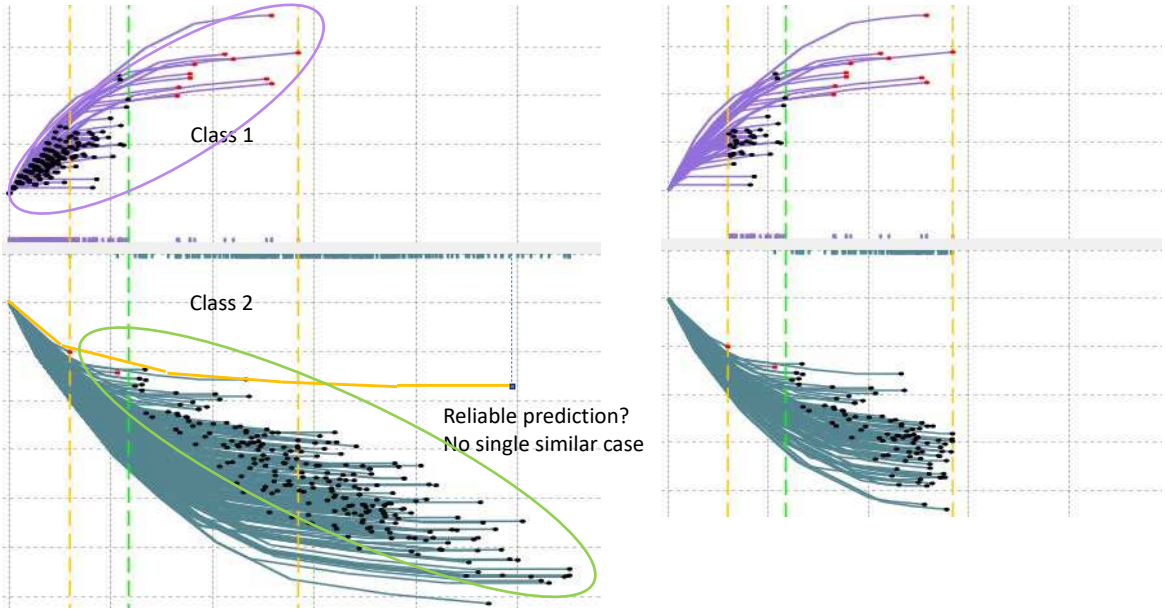
Figure 13. Learning with rejection option in General Line Coordinates-Linear (GLC-L).

Figures 13 and 14a provide another example of learning with rejections option. This is a visual method based on General Line Coordinates-Linear (GLC-L). Below we describe the idea of this method based on [Kovalerchuk, 2018].

Figure 13a shows four black coordinate lines X_1-X_4 located under different angles to the horizontal line U. The blue vectors ($\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, $\mathbf{x}_4$) located in these coordinates represent a 4-D point (x_1, x_2, x_3, x_4) = (1, 0.8, 1.2, 1). The next step shown in Figure 13b is moving the 2nd vector $\mathbf{x}_2$ to the top of the vector $\mathbf{x}_1$ and then similarly vector $\mathbf{x}_3$ to the top of the moved vector $\mathbf{x}_2$ and finally vector $\mathbf{x}_4$ is put to the top of the moved vector $\mathbf{x}_3$. Then the end point of this polyline (directed graph) is projected to the horizontal axis U.

The next step is repeating this process for several 4-D datapoints from two classes (see Figure 13 c and d) where the cases from one class are represented above U line and the cases of the second class are mirrored below this U line. The final steps are **building a discrimination vertical line** (yellow line) between the projections of case to the U line and **optimizing the location of the discrimination line** by optimizing the angles of the coordinates X_1-X_4 and moving the discrimination line accordingly. The LRO step follows by outlining acceptance areas (see Figure 13 d).

Figure 14a shows 9-D Wisconsin Breast Cancer (WBC) data [Wolberg, 1992] in GLC-L with acceptance areas on benign and malignant classes in ovals. It also shows a new yellow case that is outside of those acceptance ovals and is refused to be classified by this linear model. Note that its projection is for the green class with, “high”, confidence, if we only consider it's a location relative to the dotted green discrimination line.



(a) Controlling model overgeneralization with acceptance areas in GLC-L visualization. All WBC data, accuracy 98.1%.

(b) Worst-case validation data (overlap area), accuracy 79.74%.

Figure 14. 9-D Wisconsin Breast Cancer (WBC) data. with Linear Discriminant Function (LDF) and General Line Coordinates-Linear (GLC-L). Class malignant (green), class benign (violet) [Huber, et al., 2024].

3.3. Avoiding unreliable predictions with worst-case accuracy estimates using visual granular patterns

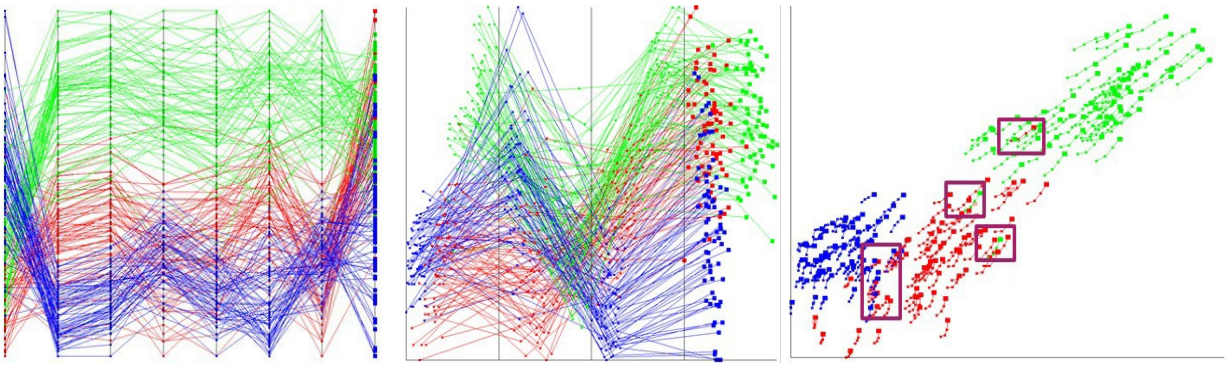
The key idea of the **visual worst-case approach** is (1) selecting interactively in lossless n-D data visualization **most difficult n-D cases** and then (2) computing the **accuracy** of different algorithms on these most difficult data when the algorithm is trained on the remaining data. We search for the algorithm, which will provide (1) the highest accuracy on such difficult data relative to other algorithms and (2) that its accuracy will be above a required level. This algorithm can be recommended to be used as most successful on difficult cases.

Figure 14a shows all WBC dataset visualized with 98.1% accuracy with GLC-L. The dotted yellow lines show the bounds for worst-case validation split. The worst-case validation set contains 22.4% of the dataset with 79.74% accuracy, which is shown in Figure 14b.

In more detail the **Worst-Case Linear** (**GLC-WCL**) algorithm is as follows [Huber, et al., 2024]:

1) Find the **lower bound** of the worst-case validation split by using the projections from GLC-L to locate the **leftmost misclassified n-D point**. If no point is misclassified set the lower bound to the threshold *T*.
2) Find the **upper bound** of the worst-case validation split by using the projections from GLC-L to locate the **rightmost misclassified n-D point**. If no point is misclassified set the upper bound to the threshold *T*.
3) If the range between the upper and lower bounds **exceeds** predefined value V% of the total range of GLC-L projections, then find another worst-case range that will be not greater than V% of the total range by excluding most extreme projection points to reach V%. **A user assigns V% value** for the task at hand.

Figure 15 shows Seeds data visualized in Parallel Coordinates (PC), Shifted Paired Coordinates (SPC), and Dynamic Scaffolding Coordinates 2 (DSC2) with worst-case split in DSC2 [Recaido Kovalerchuk, 2023]. First it demonstrates the difficulties to identify areas of heavy overlap of classes in PC and SPC. DSC2 allows users to do this much easier by outlining the overlap areas with the red rectangles shown in Figure 15c. It was done with scaling of the attributes.



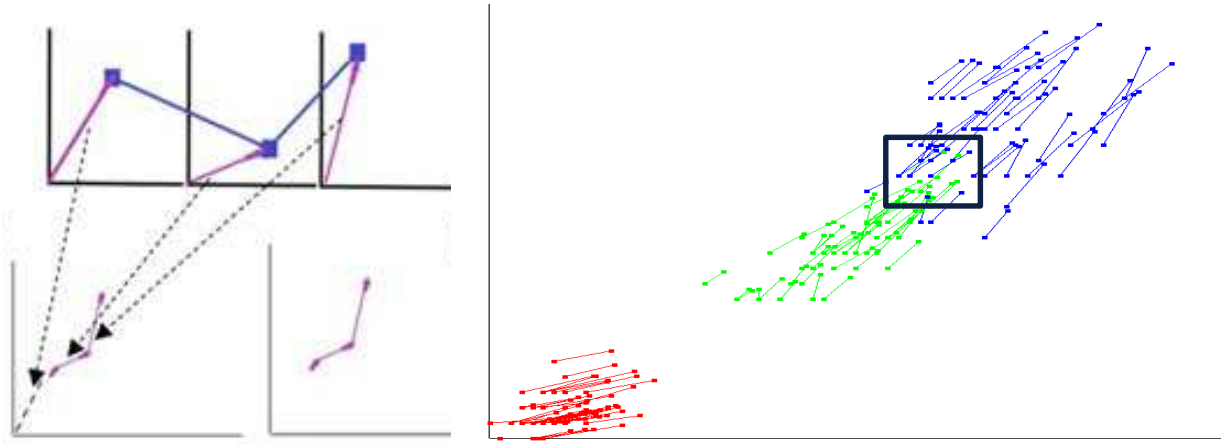
(a) Seeds on Parallel Coordinates (PC).

(b) Seeds on Shifted Paired Coordinates (SPC).

(c) Seeds Dataset Visualized on Dynamic Scaffolding Coordinates 2 (DSC2) with Attribute Scaling.

Figure 15. Seeds data visualized parallel coordinates, shifted paired coordinates and DSC2 coordinates with worst-case split in DSC2 [Recaido Kovalerchuk, 2023].

The DSC2 method is described below with an illustration in Figure 16. This is similar to GLC-L as illustrated in figures 13 and 14. The main difference is that the polyline (graph) of an n-D point is defined not by the angle of the coordinate vector as in GLC-L but the shifted paired coordinates. Figure 16a shows this process: (1) represent a 6-D point as a blue polyline in SPC, (2) generate magenta lines from SPC origins to nodes of the blue polyline, (3) build a magenta polyline by drawing magenta lines one after another (shown in by dotted black lines in Figure 16a), (4) remove the first magenta line as not necessary to have a losses representation of the n-D point in DSC2, and optional (5) scaling some pairs of coordinates allowing to **visual granular patterns** to be more visible. Figure 16b shows Iris data in DSC2 with worst-case overlap data in the black rectangle.



(a) DSC2 process for a 6-D case.

(b) Iris in DSC2 with scaling first attribute pair to 90%.

Figure 16. Dynamic Scaffold Coordinates based on Shifted Paired Coordinates (DSC2)..

Table 2 presents the results of evaluating accuracy for Seeds data. It shows the **average** of 10-Fold Cross Validation vs. the **worst-case validation** data found in Figure 15c for 7 popular machine learning algorithms: Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), k-Nearest Neighbors (kNN), Logistic regression (LR), Naïve Bayes (NB) and Multilinear Perceptron (MLP). 10-Fold Cross Validation (CV) accuracy results were 30% to 66 % higher than the worst-case split estimate. The best performing model in 10-Fold CV was Support Vector Machine (SVM) whilst the best performing model in the worst-case split estimate was Random Forest (RF).

Table 2. Worst-case split for Seeds vs. 10-fold cross validation obtained with DSC-2 visualization.

Model	10-Fold Avg. Acc.(%)	Worst-case Acc. (%)
DT	91.0	57.1
SVM	97.1	38.1
RF	91.9	61.9
KNN	88.6	23.8
LR	91.4	38.1
NB	89.0	57.1
MLP	89.5	23.81

Similar results have been obtained for Wisconsin Breast cancer (WBC) data. Table 3 presents the results of evaluating accuracy for WBC data showing the average of 10-fold cross validation vs. the worst-case validation. 10-fold CV average accuracies are 25 to 35 % higher than the worst-case split estimates. The best performing model in 10-fold cross validation is k-Nearest Neighbors (kNN) whilst the best performing model in the worst-case split estimate is Naïve Bayes (NB). The worst performing model is Stochastic Gradient Descent (SGD) with 57.60% accuracy.

Table 3. Worst-case split for WBC vs. 10-fold cross validation obtained with DSC-2 visualization.

Model	Average accuracy of 10-fold cross-validation	Accuracy of worst-case split data
DT	94.7	62.12%
SVM	97.1	62.12%
RF	96.6	65.15%
KNN	97.2	60.60%
LR	96.8	63.63%
NB	96.1	72.72%
SGD	96.2	57.58%
MLP	95.5	60.60%

These computational experiments for problem P2: Reliability of ML models in high-stakes scenarios show that often the common evaluation of accuracy ML models is **exaggerated**, which is unacceptable in applications with high cost of individual errors. The shown method to avoid unreliable predictions with exaggerated evaluation of ML model accuracy for high-stakes tasks selects an algorithm that produces the highest accuracy on the worst validation data [Kovalerchuk, 2020]. The mathematical formulation of this worst-case estimates with Shannon functions is presented in the next section based on [Kovalerchuk, 2020].

3.4. Mathematical formulation of the worst-case estimates with Shannon functions

Consider a **labeled dataset** D and a set of machine learning algorithms $\{A_j\}_{j\in J}$. Let $\{D_i\}_{i\in I}$, $I=\{1,2,\ldots,m\}$ be a **set of splits** of D to <Training data, Validation data> **pairs**. k-fold cross validation split is one of them. Each D_i is a pair of training and validation data, $D_i = (Tr_i, Val_i)$. $A_{jv}(D_i)$ is the **error rate** on validation data Val_i produced by A_j when A_j is trained on the training data Tr_i from D_i.

Split D_w is called the **worst-case split** for algorithm A_j,

$$D_w = \arg\left(\max_{i\in I} A_{jv}(D_i)\right)$$

Informally, the worst-case data split D_w among splits $\{D_i\}$ is a split, which is most difficult for the algorithm A_j. This algorithm produces max number of errors for D_w on its' validation data among all data splits from $\{D_i\}$.

Split D_b is called the **best case split** for algorithm A_j,

$$D_b = \arg\left(\min_{i\in I} A_{jv}(D_i)\right)$$

Informally, the best-case split is a split, which is easiest for the algorithm A_j, which produces fewer errors on the validation data than the other algorithms on their splits from $\{D_i\}$.

Split D_m is called the **median split** *for algorithm* A_j,

$$D_m = \arg\left(\underset{i \in I}{\text{median}}(A_{jv}(D_i))\right)$$

Informally, the median split for the algorithm A_j produces the error rate that is close to the average error rate among $\{D_i\}$ for A_j algorithm. The worst and best estimates compliment traditional expectation estimate like median split providing upper and bottom estimates of the expected errors**.**

The adaptation of the **Shannon function** $S(I, J)$ to supervised learning problem is defined as follows:

$$S(I,J) = \min_{j \in J} \max_{i \in I} A_{jv}(D_i) \tag{4}$$

It outputs the error rate of the algorithm that generates a smallest error rate on its worst data split D_i, among all considered algorithms. Respectively, the algorithm A_b is called ***S*-best algorithm (Shannon best algorithm)** if its error rates $A_{bv}(D_i)$ satisfies (1):

$$S(I,J) = \min_{i \in I} A_{bv}(D_i) \tag{5}$$

In other words, the Shannon best algorithm produces **fewer errors** on validation data on its **worst splits** among $\{D_i\}$ than other algorithms on their worst splits among $\{D_i\}$. In practice getting exact values $S(I, J)$ can be difficult and low and upper bounds for worst-case estimates can be used. The presented above examples show the ability of finding worst-case validation sets where their performance is much worse than average performance in traditionally used k-fold cross validation.

4. Conclusion and Future work

Explanations of models are mandatory for high-stakes machine learning tasks, which range from medical diagnosis financial decision-making, autonomous vehicles, criminal justice to disaster response planning, search and rescue operation, and others. Explanation must accurately describe the model's inner workings convincing the user that it is acceptable to apply the model. Many current popular AI/ML explanation methods do not reach this level being only quasi-explanations, but some black-box models have been deployed without proper explanation increasing risk of wrong decisions.

This study analyzed current methods with their benefits and deficiencies as well as ways to overcome their deficiencies with focus on high-stakes AI/ML tasks to get *reliable* and *trustworthy* models. It is presented from the viewpoint of the major topics of emerging importance in interpretable AI/ML models for high-stake tasks such as (1) *methodology* of explaining black-box models, (2) *computational methods* to explain AI/ML models, (3) *human-in-the-loop*, and (4) *Visual Knowledge Discovery* (VKD).

We discussed the issue: "What went wrong and why: lessons from AI research and Applications?" raised at the AI conference at Stanford celebrating 50 years of AI in 2006 along with the current question: "What is failing now in AI?" The declared DARPA goal of explainable AI (XAI) program to produce explainable models is still a goal.

This study focused on two open problems: P1) Explanation of ML models in High-stakes Scenarios and P2) Reliability of ML models in High-stakes Scenarios. Table 1 summarized the properties and requirements for the high-stakes tasks vs. low-stakes tasks. For problem P1 we have shown that many explanations are rather quasi-explanation than real ones. Next, we presented explanation methods for ML models beyond quasi-explanation including ones based on visual knowledge discovery paradigm. For problem P2 we have shown that often common evaluation of accuracy ML models is exaggerated, which is unacceptable in applications with high cost of individual errors. Then we presented the methods that avoid unreliable predictions with exaggerated evaluation of ML model accuracy for high-stakes tasks. It is based on the combination of the worst-case estimate with the Shannon function and the visual knowledge discovery paradigm. The implementations of the presented approaches for P1 and P2 in software are available at GitHub in [VKD, 2024].The future work will go to expanding the presented approaches with a wider set of analytical and visual knowledge discovery methods in combination with multiple emerging machine learning methods.

5. Declaration

Author contribution: Boris Kovalerchuk is a sole contributor to this study.
The author declares no conflict of interest.
No funding was obtained for this study.
Data Availability Statement: no data generated in this study outside the manuscript file.

6. References

1. Albahri S, Duhaim AM, Fadhel MA, Alnoor A, Baqer NS, Alzubaidi L, Albahri OS, Alamoodi AH, Bai J, Salhi A, Santamaría J. A systematic review of trustworthy and explainable artificial intelligence in healthcare: Assessment of quality, bias risk, and data fusion. Information Fusion. 2023 Mar 15.
2. Bartlett PL, Wegkamp MH. Classification with a Reject Option using a Hinge Loss. Journal of Machine Learning Research. 2008 Aug 1;9(8).
3. Big Data and Machine Learning, 2018, http://www.cnblogs.com/luweiseu/p/7826679.html
4. Craik KJ. The nature of explanation. CUP Archive; 1967.
5. DARPA XAI program, 2016, https://www.darpa.mil/program/explainable-artificial-intelligence.
6. Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608. 2017 Feb 28.
7. Freiesleben T, Grote T. Beyond generalization: a theory of robustness in machine learning. Synthese, 2023 Sep 27;202(4):109.
8. Fryer D, Strümke I, Nguyen H. Shapley values for feature selection: The good, the bad, and the axioms. IEEE Access. 2021 ;9:144352-60.
9. Gao J, Yao J, Shao Y. Towards reliable learning for high stakes applications. In: Proceedings of the AAAI Conference on Artificial Intelligence 2019 Jul 17 (Vol. 33, No. 01, pp. 3614-3621).
10. Ghassemi M, Oakden-Rayner L, Beam AL. The false hope of current approaches to explainable artificial intelligence in health care. The Lancet Digital Health. 2021 Nov 1;3(11):e745-50.
11. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.
12. Gori M, Betti A, Melacci S. Machine Learning: A constraint-based approach. Elsevier; 2023 Mar 1.
13. Hein M, Andriushchenko M, Bitterwolf J. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019 (pp. 41-50).
14. Hayes D., Kovalerchuk, B., Parallel Coordinates for Discovery of Interpretable Machine Learning Models, In: Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery, pp. 125-158, Springer, 2024. https://arxiv.org/pdf/2305.18434.
15. Huber L., Kovalerchuk B., Recaido C., Visual Knowledge Discovery with General Line Coordinates, In: In: Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery, pp. 159-202, Springer, 2024. https://arxiv.org/pdf/2305.18429.
16. Huang Z, Wang H, Huang D, Lee YJ, Xing EP. The two dimensions of worst-case training and their integrated effect for out-of-domain generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022, pp. 9631-9641.
17. Kovalerchuk B., Visual Knowledge Discovery and Machine Learning, Springer Nature, 2018
18. Kovalerchuk B, Grishin V. Reversible data visualization to support machine learning. In: Human Interface and the Management of Information. Interaction, Visualization, and Analytics. Proceedings, Part I 20 2018, pp. 45-59, Springer.
19. Kovalerchuk B. Enhancement of cross validation using hybrid visual and analytical means with Shannon function. In: Beyond Traditional Probabilistic Data Processing Techniques: Interval, Fuzzy etc. Methods and Their Applications. 2020, pp. 517-543, Springer.
20. Kovalerchuk B, Nazemi K, Andonie R, Datia N, Banissi E, (Eds). Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery, Springer, 2022
21. Kovalerchuk B, Ahmad MA, Teredesai A. Survey of explainable machine learning with visual and granular methods beyond quasi-explanations. Interpretable artificial intelligence: A perspective of granular computing. pp:217-267, Springer, 2021. https://arxiv.org/pdf/2009.10221
22. Kovalerchuk B. Explainable machine learning and visual knowledge discovery. In: Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook 2023, pp. 913-943. Springer.
23. Kovalerchuk, B., Dunn A., Worland A., Wagle S., Interactive Decision Tree Creation and Enhancement with

Complete Visualization for Explainable Modeling, In: Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery, pp. 3-40, Springer, 2024. https://arxiv.org/pdf/2305.18432.
24. Kovalerchuk B, Fegley BD. Principal Components in General Line Coordinates for Visualization and Machine Learning. In: 2023 27th International Conference Information Visualisation (IV) 2023, pp. 300-307. IEEE.
25. Kovalerchuk B. Quest for rigorous intelligent tutoring systems under uncertainty: Computing with Words and Images. In: 2013 Joint IFSA World Congress and NAFIPS Annual Meeting 2013, 685-690. IEEE.
26. Kumar IE, Venkatasubramanian S, Scheidegger C, Friedler S. Problems with Shapley-value-based explanations as feature importance measures. In: International Conference on Machine Learning 2020, 5491-5500). PMLR.
27. Lakkaraju H, Bastani O. " How do I fool you?" Manipulating User Trust via Misleading Black Box Explanations. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society 2020, pp. 79-85.
28. Lei J, Li D, Xu C, Fang L, Hospedales TM, Fu Y. Worst-case Feature Risk Minimization for Data-Efficient Learning. Transactions on Machine Learning Research. 2023 Oct 26;2023(09):1-9.
29. Lin Wan-Yi, When a stop sign turns into a vase — Robust machine learning under worst-case perturbations, 2023-07-20, https://www.bosch.com/stories/robust-machine-learning/
30. Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, Katz R, Himmelfarb J, Bansal N, Lee SI. Explainable AI for trees: From local explanations to global understanding. arXiv preprint arXiv:1905.04610. 2019 May 11.
31. Lundberg SM, Lee SI. A Unified Approach to Interpreting Model Predictions. In: Advances in Neural Information Processing Systems 30. 2017, 4768–4777. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.
32. Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S. I. (2020). From Local Explanations to Global Understanding with Explainable AI for Trees. Nature machine intelligence, 2(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9
33. Mayo, D. G., & Spanos, A. (2006). Severe testing as a basic concept in a Neyman–Pearson philosophy of induction. The British Journal for the Philosophy of Science, 57(2), 323–357.
34. Molnar. C. 2020. Interpretable Machine Learning. https://christophm.github.io/interpretable-ml-book/.
35. Recaido C, Kovalerchuk B. Visual Explainable Machine Learning for High-Stakes Decision-Making with Worst Case Estimates. In: Data Analysis and Optimization, 2023, pp. 291-329. Springer Nature Switzerland.
36. Michel P, Ruder S, Yogatama D. Balancing average and worst-case accuracy in multitask learning. arXiv preprint arXiv:2110.05838. 2021 Oct 12.
37. Raschka, S. MLxtend, 2023. https://rasbt.github.io/mlxtend/
38. Rizzo M, Veneri A, Albarelli A, Lucchese C, Nobile M, Conati C. A Theoretical Framework for AI Models Explainability with Application in Biomedicine. 2023, https://arxiv.org/pdf/2212.14447
39. Robey A, Chamon L, Pappas GJ, Hassani H. Probabilistically robust learning: Balancing average and worst-case performance. In: International Conference on Machine Learning 2022, 18667-18686. PMLR.
40. Rudin C. Why black box machine learning should be avoided for high-stakes decisions, in brief. Nature Reviews Methods Primers. 2022 Oct 27;2(1):81.
41. Sahoh B, Choksuriwong A. The role of explainable Artificial Intelligence in high-stakes decision-making systems: a systematic review. Journal of Ambient Intelligence and Humanized Computing. 2023 Jun;14(6):7827-43.
42. Theisen RC. Beyond Worst-Case Generalization in Modern Machine Learning, Doctoral dissertation, University of California, Berkeley, 2023, https://escholarship.org/uc/item/62j5g7vd.
43. VKD, CWU Visual Knowledge Discovery Lab software, 2024, https://github.com/CWU-VKD-LAB,
44. Watson DS, Conceptual challenges for interpretable machine learning, Synthese (2022) 200:65, https://link.springer.com/content/pdf/10.1007/s11229-022-03485-5.pdf
45. Wolberg W., Breast Cancer Wisconsin (Original), 1992, https://archive.ics.uci.edu/dataset/15/breast+cancer+wisconsin+original
46. Worland A, Wagle S, Kovalerchuk B. Visualization of Decision Trees based on General Line Coordinates to Support Explainable Models. In: 2022 26th International Conference Information Visualisation (IV) 2022, pp. 351-358. IEEE.
47. Yang K, Lin WY, Barman M, Condessa F, Kolter Z. Defending multimodal fusion models against single-source adversaries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021 (pp. 3340-3349).
48. Zhang XY, Xie GS, Li X, Mei T, Liu CL. A survey on learning to reject. Proceedings of the IEEE. 2023 Jan 27;111(2):185-215.