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. For high-stakes tasks, effects of incorrect decisions or predictions can cause significant harm to individuals or society. This paper presents (1) major topics of research in interpretable artificial intelligence and machine learning (AI/ML) for high-stakes tasks with a human-in-the-loop, and (2) motivates and addresses 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 true goal of model explanation. This paper presents both benefits and deficiencies of current methods and ways to overcome said 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, 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 this 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, model 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 vehicle navigation* (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 [Rudin, 2019; Sahoh, Choksuriwong, 2023].

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* and *convincing* to the user [Rizzo, et al., 2023]. If these properties are 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.

In contrast to the concept of ML model *accuracy*, the concept of a *convincing* ML model fundamentally depends on a *human* who needs to be convinced that said model is acceptable [Adebayo, et al., 2018; Gupta, et al., 2002, Zhang, et al., 2018]. The ideal situation for defining the **convincing** ML model would be if (1) a human **mental model** or **domain model** is available and (2) is **mapped** to the ML model in a formalized and **justified** way. Unfortunately, in many domains with high-stakes decisions, like self-driving cars or medicine, we lack (1) and/or (2) at required levels.

Satisfying (1) and (2) would constitute a **real**/**trusted**/**normative explanation** whether that model behavior was justified, in contrast with a **descriptive explanation**, which only describes model's behavior. While most

discussions and policies call for normative evaluations, current techniques are only capable of descriptive accounts [Ghassemi, et al., 2021].

Therefore, today the decision that the ML model is convincing is derived from interviewing humans. Moreover, the performance of explanations is rarely tested at all, and most tests that are done rely on heuristic measures rather than explicitly scoring the explanation from a human perspective [Ghassemi, et al., 2021].

The ideal situation of the user's mental domain model mapped with the explanation could allow for **measuring the benefits** of different explanation techniques, including the following visualization techniques with decision trees. Unfortunately, as we pointed out above, the current interpretability progress is far away from this ideal situation to be able to be widely used. Another option to measure benefits is the measuring the level of **removal of foreign terms and assumptions** from the explanations in the form of user-study, which will confirm that all terms and assumptions are verifiable and acceptable for the user's domain. Unfortunately, this approach is also in a nascent stage.

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/trusted* models. The success of 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 [Kovalerchuk, et al., 2021]. Therefore, a significant part of this paper is devoted to discovering interpretable ML models with human visual knowledge discovery methods.

The analysis of a model's inner reasoning is conducted by: (1) **data scientists** who need to *debug* a model to eliminate errors and to make it more accurate, and (2) **domain experts/end users** who need to *use* this model being convinced that the model makes sense in the application domain [Capel, Brereton, 2023; Hohman, et al., 2019; Zytek, et al., 2021]. The needs (1) and (2) are significantly different and each respectively leads to *very different requirements* for analysis of inner reasoning of the model. In general, we need both the *system's ability* to explain decisions or predictions and the *human's ability* to interpret this system's explanation within the **Humal-Centered AI** (HCAI) [Capel, Brereton, 2023, Shneiderman, 2022].

Consider a simple linear classification model that classifies 2-D cases $\mathbf{x} = (x_1, x_2)$, if $2x_1 + 3x_2 < 10$ then $\mathbf{x}$ belongs to class 1, else $\mathbf{x}$ belongs to class 2. The inner reasoning of this model is just computing $2x_1 + 3x_2$ and comparing it with 10. For $\mathbf{x} = (2, 1)$ we get $4 + 3 < 10$ and predict that the case belongs to class 1. If this prediction is incorrect, the data scientists knowing this inner reasoning (mathematical structure of this linear function) tries to adjust coefficients and thresholds to correctly classify this case, controlling that other training and test cases are not misclassified.

This reasoning via the mathematical structure of the function does not explain much to the domain expert, especially if attributes x_1 and x_2 are heterogeneous, like blood pressure and pulse. The weighted summation of such attributes has no meaning in the medical domain as we discuss in sections 1.3 and 2.1, which leads to a **quasi-explanation**. More complex models like deep neural network use multiple such linear functions, with many weights and thresholds. Presenting details of these complex computational processes to domain experts does not help them much. Often these details are "**foreign**" to the user domain, without direct meaning there, and not expressed in the domain terms [Kovalerchuk, et al., 2021]. Moreover, complexity of those computations makes analysis of these computations non-trivial for the data scientists too. There is a risky proposition that domain experts need to use accurate black-box models *despite absence of satisfactory explanation* of them. Following such a proposition would require tremendous demonstration of success of these models, and each cumulative failure will diminish their acceptance.

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, unjust punishment, and others.

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-stakes 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 extremely 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 interpretation. 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.

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 (1) for explanations as an **interactive dialogue** where end users, who can query or explore different explanations, (called perspectives) of the black-box, and (2) for capturing **causal relationships** between input features and black-box predictions, because relying on correlations can be misleading and unrobust. The V**isual Knowledge Discovery** approach is fully within this paradigm. We advocate for interaction in lossless n-D data visualizations, along with visualizations of ML models, as a natural and efficient way of interactive dialogue, detailed in section 2.5.

Development of new robust ML systems for safety-critical applications is urgently needed for testing in the **most unfavorable conditions** to ensure that they continue to work *correctly* in a potential 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 (undergeneralization)** and data **underfitting** (**overgeneralization**) [Theisen, 2023], which are especially challenging under distribution shifted, low-data availability, and imbalanced data distribution situations. 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) robustness to **adversarial small changes** to input data, e.g., [Yang, et al., 2021] and (2) **minimax optimization**. The latter includes our approach with the **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.

This paper is organized as follows. Section 1.1. provides overview of the topics covered. Section 1.2 outlines the concept of Visual Knowledge Discovery as one of the key frameworks for dealing with high-stakes tasks. Section 1.3 puts current interpretability challenges in the historic context of the AI field. Section 1.4 presents the concept of

explanation-driven interactive machine learning (XIML). Section 1.5 describes the paper scope, outlines main concepts, and provides core statements about them. Section 2 is devoted to the 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 the problem P2 (reliability of ML models in high-stakes scenarios). In section 3.1 we show that often the common evaluation of ML model accuracy 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-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. The references section presents multiple relevant publications. Readers further interested in these publications can search Google Scholar for downloading them.

1.2. Visual Knowledge Discovery

Visual Knowledge Discovery (**VKD**) is a process of visualizing and exploring complex high-dimensional information [Kovalerchuk, et al., 2018-2024]. 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.

Visual Knowledge Discovery needs a **lossless visualization** of n-D data, but it is impossible in popular visualization methods like PCA and t-SNE which map an individual n-D point to an individual 2-D or 3-D point. The Johnson-Lindenstrauss lemma [1984] clearly shows this deficiency [Kovalerchuk, 2018]. In contrast, the methods that map each n-D point to a **graph** in 2-D or 3-D are lossless]. It is possible because the graph is a rich structure in contrast with a single 2-D or 3-D point [Kovalerchuk, 2018], which is illustrated in section 2.

Respectively, a **lossless/reversible visualization** is defined as a visualization that allows for *restoration* of all n-D data, transforming visualization back to data. A similar property is known for lossless image/data compression, which allows a full restoration of the image/data from its compressed form. Parallel and radial coordinates are lossless visualization methods which have been known for a long time [Inselberg, Dimsdale, 2009]. Their extension to the **General Line Coordinates** (**GLC**) dramatically expanded the number of these methods [Kovalerchuk, et al., 2018-2024]. Several such methods are demonstrated in section 2, including *Shifted Paired Coordinates* (SPC), *GLC-Linear* (GLC-L), and *Bended Coordinates*.

Visual Knowledge Discovery has a broader appeal than visualization, it includes discovering new knowledge using visual means beyond visualizing already discovered entities like data, information, and knowledge. The difference between these two ways is illustrated in the following example with data from Table 1.

Table 1. Data example.

x	1.3	2.2	7.1	5.9	11	2.8	10.5	4.3	6.4
y	2	4	13	12	20.3	5.4	22.1	8.2	12.2

In this example the first way (**visualization way**) is:
(1.1) **Discovering computationally** an approximation formula of y by x, $F(x) = 2x = y$ `.
(1.2) *Visualizing this already discovered formula* $F(x) = 2x$, as shown in Fig. 1 by a red line.
It does not require to visualize data pairs (x, y) from Table 1, but it is desirable to see the accuracy of the approximation.
The second way (**Visual Knowledge Discovery way**) is:

(2.1) *Visualizing* data pairs (x, y) from Table 1.
(2.2) **Discovering visually and analytically** the linear pattern.
(2.3) *Drawing* a red line and *finding* its mathematical form $F(x) = 2x$ from its drawing in Figure 1.

The second way is a quite common visual discovery method for *one-dimensional patterns,* but it is **not feasible** for high-dimensional data. The standard Cartesian coordinates used in Fig.1 for this visualization are limited to 2-D/3-D data, while in Machine Learning we deal with dozens and hundreds of dimensions. Fundamentally **lossless visualization methods** are required to visualize multidimensional (n-D) data preserving all high-dimensional information to conduct visual knowledge discovery in n-D data. Figure 2 contrasts visualization and visual discovery.

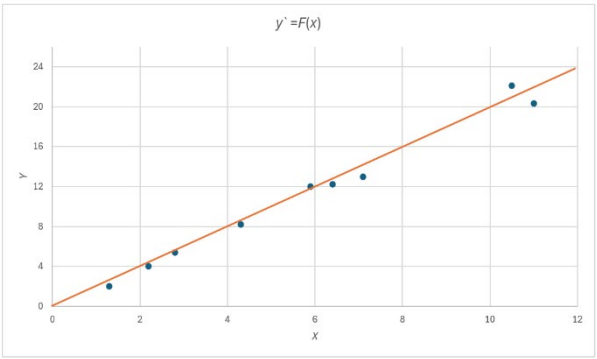


Figure 1. Illustration of visual discovery of function $F(x)$ vs. visualization already discovered function $F(x)$.

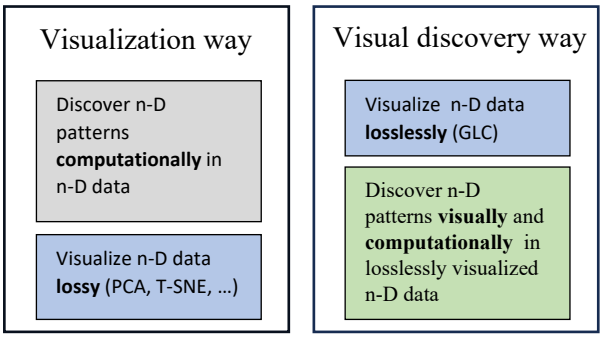
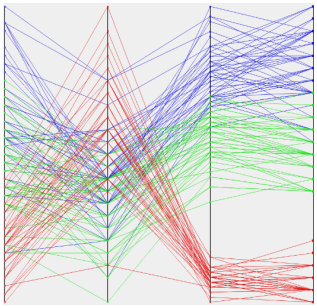
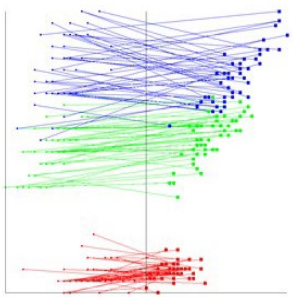


Figure 2. Contrasting visualization and visual discovery.

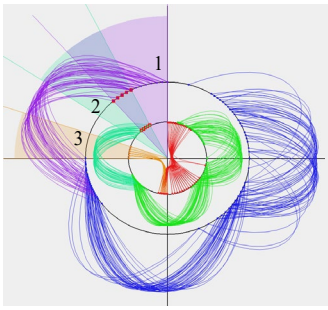
VKD methods rely on **lossless General Line Coordinates** (**GLC**) [Kovalerchuk, et al., 2018-2024], which generalizes the Cartesian coordinates. In contrast with the classical orthogonal 2-D/3-D Cartesian coordinates used to visualize 2-D/3-D data, GLC allow any number of coordinates located in 2-D/3-D space in any direction and location. GLC can be parallel, non-parallel, starting from the common origin or from multiple origins, they can overlap, be located in a single line, be shifted relative to each other, be straight lines or curves, can form open or closed shapes such as polylines, circles, ellipses, squares, pentagons and any n-dimensional polygon. Figure 3 shows some of them with details in section 5.3.



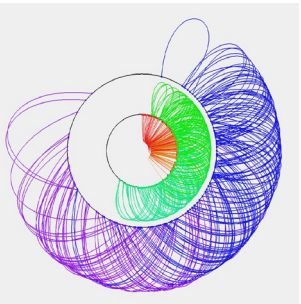
(a) Parallel Coordinates (PC)



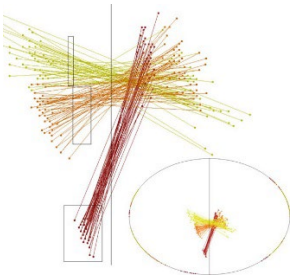
(b) Shifted Paired Coordinates (SPC)



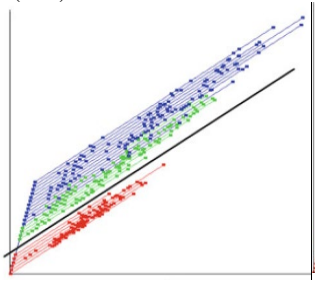
(c) Static Circular Coordinates (SCC).



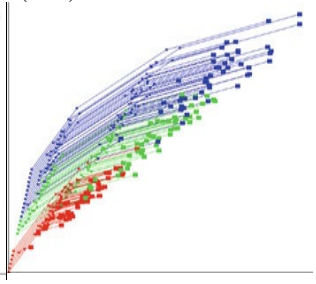
(d) Dynamic Circular Coordinates (DCC).



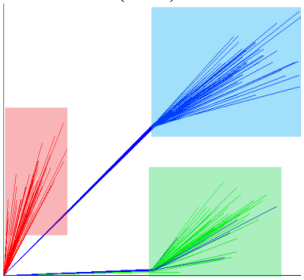
(e) Elliptic Paired Coordinates (EPC)



(f) Dynamic scafolding coordiantes 1 (DSC1) linear



(g) Dynamic scafolding coordiantes 1 (DSC1) non-linear



(h) Dynamic scafolding coordiantes 2 (DSC2)

Figure 3. Iris data in several forms of GLC [Recaido et al, 2023; Williams et al, 2024; McDonald et al, 2022; Kovalerchuk et al, 2018-2024].

Fig. 3a shows Iris data in the well-known Parallel Coordinates (PC). Fig 3b show **Shifted Paired Coordinates** (**SPC**), where each 4-D point **x**=(x_1,x_2,x_3,x_4) is mapped to a line that connects pair (x_1,x_2) with pair (x_3,x_4). Each pair is shown in its respective Cartesian coordinates: (x_1, x_2) in (X_1, X_2) and (x_3, x_4) in (X_3, X_4). The color of lines indicates the class cases belong to in the data, here being of Iris flower species. Each line is a **lossless** representation of a 4-D point **x**. This lossless visualization allows to discriminate classes visually by visually selecting point containing rectangles, shown in Figure 3a and to see overlap cases shown in Figure 3b. To represent losslessly data with more than 4 dimensions, SPC uses multiple Cartesian coordinates, pairing attributes at points and connecting corresponding points with lines, explained in section 2.5.3. For data with an odd number of attributes the last coordinate is duplicated.

1.3. 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 Spring 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 to AI development, but also of the 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 the coining of this term. This field reached its 50 years in 2015. Its competition with DNNs is not obvious.

Discussions on both AI approaches based either on the classical logic or the 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 the 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 just 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 possibly reach beyond quasi-explanations [Kovalerchuk, et

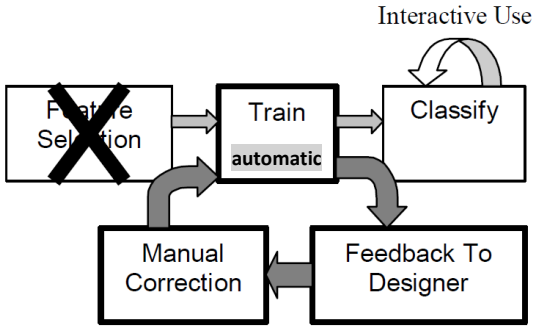
al., 2021] for difficult high-stakes tasks, situations where users must appropriately trust models. The development of an effective explanation interface was also one of the goals of DARPA XAI program with, "the appropriate use of **visualization** and natural language understanding," and, "dialogue 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 analyzed here are within this Human-Centered AI (HCAI) paradigm.

1.4. Explanation-driven interactive machine learning (XIML)

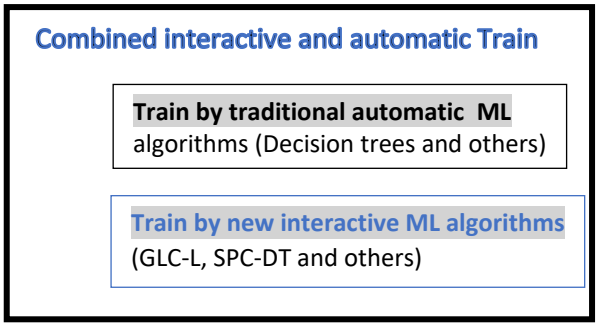
The survey of **Human Centered AI** (**HCAI**) [Capel, Brereton, 2023] identified its major areas as Explainable and Interpretable AI, Human-Centered Approaches to Design and Evaluate AI, Humans Teaming with AI, and Ethical AI, with a newer area, Interaction with AI that is at the intersection of four major areas. Our focus is on *human in-the-loop approaches in Explainable and Interpretable AI/ML* to benefit both data scientists and application domain experts. This topic is closely related to the issues of Trustworthiness and Transparency of AI/ML models. The challenge is that explanations are very specific to individual people requiring 'tailorability' of explanations for the human-in-the-loop (HITL), which is analyzed in [Capel, Brereton, 2023] with multiple references. We focus on **human-in-the-loop** in the form of **Visual Knowledge Discovery**, where humans can be naturally involved in the *AI pipeline* of model creation and use; through labeling data, discovering patterns, training the model, correcting misclassification, evaluating, tuning, explaining the model, and actual use of the model.

Explanation-driven interactive machine learning (XIML) seeks to leverage user feedback to iterate on an ML solution *to correct errors* and *align decisions* with users' feedback on *explanations* [Guo, et al., 2022]. This formulation is fully in the **traditional ML paradigm** to **iterate** on an ML solution by applying *automatic* ML algorithms multiple times until an acceptable solution will be discovered. The AutoML approach [He, et al., 2021] focuses on *automatic iteration* and XILM focused on *interactive iteration*. However, this interaction is only **before** and **after** the automatic stage of actual model discovery, **not during** it when the automatic algorithms like Random Forest, SVM, Neural Networks and others run. This means that this type of interaction is rather *auxiliary* or *supporting* machine learning, rather than being at the core of it. It is pointed out in [Fails, Olsen, 2003] that to conduct interactive ML with automatic ML algorithms they must run fast enough to support interaction. A similar situation is with *interactive explanation* processes where the automatic explanation algorithms like LIME are augmented with interactive inputs to improve explanations.

A framework such as **Interactive Machine Learning** (**IML**) was, likely, first articulated in [Fails, Olsen, 2003]. IML suggests a preliminary selection of a large pool of available features, and then using an automatic ML algorithm (depicted in Figure 4a as "Train automatic") iteratively after each manual correction of inputs. It is illustrated with the Crayons system in that paper, where a user roughly outlines the objects of interest manually on the images, then an automatic ML algorithm produces a model. The user corrects inputs by providing counterexamples to fine tune this classifier to decrease the overgeneralization of the classes. IML can be considered as a move from relatively passive learning to more active interactive learning [Teso, Kersting, 2019], which is not complete as we pointed out above.



(a) Interactive ML with human passive automatic training by traditional ML algorithms (decision trees and others) based on [Fails, Olsen, 2003].



(b) Human active combined training by interactive ML algorithms and traditional automatic ML algorithms.

Figure 4. Human Passive vs. Human Active Interactive ML training.

Several XIML systems have been developed within the traditional ML paradigm, which use automatic ML algorithms. Cuo, et al., [2022] developed an XIML system with the Tic-Tac-Toe game as a use case to understand how an XIML mechanism improves users' satisfaction with the machine learning system. The authors report the study increased user satisfaction. In [Slany, et al., 2024] model-agnostic XIML system allows physicians to iteratively train the ML model and revise its decision-making mechanism depicted as local explanation. Counter examples serve as additional training data with probabilistic rule-based additions to correct a model's decision process. The interactive rule-based addition is one of the ways to expand XIML paradigm to a more interactive process of the actual model discovery.

In [Teso, et al., 2023] a survey of the topic is provided, called there, a general approach for leveraging explanations in interaction. It lists and reviews over 20 relevant methods. This approach is illustrated with the EluciDebug system [Kulesza, et al., 2015], which customizes interactively Naïve Bayes classifiers for email classification. A user gets an explanation as relative contributions of words that the model uses to distinguish work emails from personal emails. The user can (1) change the relevance of certain input variables by adjusting the model weights, and (2) specify relevant words that the system ignores and irrelevant words that the system is wrongly uses. This interaction is more specific than traditional label-based strategies. In the label-based strategies a user changes the training data but has no control that the system will actually use it, e.g., tells the system that a specific message is personal rather than work-related. In EluciDebug system the interaction continues until a user is satisfied with the system's behavior. It is beneficial for the users as reported in [Kulesza, et al., 2015], however conceptually it still uses the traditional automatic Naïve Bayes classifier, with more active changes of its input.

Another survey [Finzel, 2024] shows other examples of interactive ML with explanations. One of them is *LearnWithME* [Schmid, Finzel, 2020] for digital pathology with colon tissue. Medical experts can *correct labels* of classified examples, *explanations* and add *user-defined constraints* for adaptation of the Inductive Logic Programming algorithm learning process. The next example in CAIPI approach [Teso, Kersting, 2019] that employs explanation corrections as an additional feedback channel in a model and explainer agnostic fashion. All these user's activities are with the inputs of the existing ML algorithms, or the existing ML explanation algorithms.

We advocate an **extended paradigm** where new **interactive ML algorithms** (**IMLA**) augment traditional automatic ML algorithms with human interaction at all three stages **before, during** and **after** the model is discovered and explained. Figure 4b depicts the extension of the training block from Figure 4a to combine interactive and automatic training stages. Interaction during the actual model discovery is important, because commonly interactions before and after are to *fix deficiencies of the model discovery stage* as Figure 4a illustrates. Thus, it can help to make interaction before and after more efficient too. Interaction stages have different challenges and require different forms of interaction to resolve them.

It is **not accidental** that interaction at the actual model discovery training/stage is missing in the current dominant XILM framework is extremely challenging. This stage is fundamentally rooted in the nature of ML as a process of knowledge discovery in **multidimensional data**. To interact with these multidimensional data at the discovery stage, we need to see these multidimensional data, to be able to manipulate with them and construct a model. As we point out above, unfortunately, most of the available methods to operate interactively and visually with multidimensional data are **lossy,** like principal component analysis, multidimensional scaling, t-SNE, and others. These methods do not preserve all multidimensional information, making it practically impossible to discover the same models in these abridged data, as in the full data. **Visual Knowledge Discovery in multidimensional spaces** was proposed to resolve this issue based on a new concept of the **lossless General Line Coordinates**, which is described in section 1.2.

A framework of **explanatory interactive machine learning** (EIML) process is proposed in [Teso, Kersting, 2019]. Below is our reformulation of it, for easier analysis:

(1) A selected ML algorithm *A* produces an ML model *M* on data *D*.
(2) A user generates a query to get an explanation of the prediction by model *M* of some cases.
(3) A selected explanation algorithm *E* is applied to *M* to explain predictions of the model *M* for these cases.
(4) A user interactively evaluates the explanation by putting it to some category.
(5) A user corrects the explanation if needed by offering counter examples.
(6) Counter examples are fed to the ML algorithm *A* that produces a corrected model *M`*.
(7) A corrected explanation by explantion method *E* is produced.
(8) Stages (2)-(4) are repeated until a satisfactory for the user ML model and its explanation are generated.

While there is no doubt that this is a valuable and logical framework, however its efficiency heavily depends on the selected ML and explanation algorithms. If the ML algorithm produces explainable models, then a separate explanation algorithm may not be required. Otherwise, the explanation algorithm *E* will be required. Next, if the selected explanation algorithm *E* is a quasi-explanation algorithm, then correction may require significant effort with multiple repetitions of stage (8), which may include a complete rejection of the explanation. In [Teso, Kersting, 2019] the computational experiment was conducted by using LIME, where the coefficients of the LIME linear function are interpreted as the importance of the attributes. LIME is a quasi-explanation algorithm for ML tasks with heterogeneous data as we discuss in section 2.1, which increased chances that its explanation will be rejected by the user.

Another issue is that on stage (6) the ML system produces an alternative model **automatically** without a user in the loop. A user is not involved in actual correcting of the ML model, but instead only provides counter examples through additional input training data. This is the same issue as we discussed above for the Explanation-driven interactive machine learning framework. The framework proposed in [Teso, Kersting, 2019] is also fundamentally in the *traditional machine learning paradigm* where the existing machine learning algorithm *A* automatically produces the model *M* for the given data *D*. A more complete framework would need to modify stage (6) making it interactive:

(6`) Counter examples are fed to the **interactive ML algorithm** *A* that produces a corrected model *M`* like shown in Figure 4b.

As previously mentioned, the Visual Knowledge Discovery methodology allows for constructive addressing of this issue, where the domain experts/end users can *actually build ML models interactively*, not only support automatic model discovery as conducted by traditional automatic ML algorithms. This interactive process can produce an interpretable model *without the need in an additional interpretation method*. It allows to fundamentally expand the user involvement beyond producing the counter examples to interactive direct the model correction and interactive discovery of new models directly from multidimensional data.

1.5.Scope, definitions and core statements

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 2 illustrates characteristics of such tasks.

Table 2. 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	Often smaller	Often larger
3	Imbalance of classes in training data	Often higher	Often lower
4	Model overgeneralization	Often not acceptable	Often acceptable
5	Model overfitting	Often not acceptable	Often acceptable
6	Model accuracy by average number of errors	Often not acceptable	Often acceptable
7	Model accuracy by worst number of errors	Often required	Often not required
8	Explanation of model	Often required	Often not required
9	Low quality of model explanation methods	Often not acceptable	Often acceptable
10	Current quality of model explanation methods	Often insufficient	Often sufficient
11	Extensive domain expert knowledge to accept model	Often required	Often not required
12	Use of model	Often second opinion	Often primary decision
13	Current level of model use	Often experimental	Often industrial scale

Several studies have already been conducted in this area. The focus of this study is to **integrate** them together with a combination of Visual Knowledge Discovery and Worst-case estimates for **reliable and interpretable machine learning**. Many terms and their meaning in the AI/ML interpretability domain are not fully settled. To avoid possible misinterpretation of terms Tables 3 and 4 describe main terms, and how they are used in this paper, along with core statements about these concepts. These tables can also guide a reader in navigating this paper to find specific topics.

Table 3. Outline of main general concepts with core statements.

Concept	Description and core statements	Section
High-stakes AI/ML tasks	AI/ML tasks with high cost of individual errors.	1.1
Traditional **computational** ML algorithms: SVM, decision trees, random forest, Neural Network, etc.	Algorithms that *purely computationally* (automatically) classify every n-D point of the feature (attribute) space often utilize *implicit* and *"foreign" assumptions* for the domain experts, e.g., all n-D points of the feature space are *legitimate cases* in the domain. It often leads to data *overgeneralization* (*underfitting*), which is risky for high-stakes tasks.	2.2
Interactive Machine Learning algorithms (IMLA)	Extended ML algorithms that augment traditional automatic ML algorithms with human interaction *before, during,* and *after* the model discovery/training with explanation to improve them. In the traditional framework a user interacts only before and after training of the automatic ML algorithms. Interaction **during** discovery allows for avoiding errors that before and after interactions try to fix *externally*.	1.4
Machine Learning algorithms with **Reject** Option (LRO)	Extended ML algorithms that augment traditional ML allowing to reject to classify some cases. It counters data *overgeneralization* by traditional ML models, which is critical for high-stakes tasks.	3.2
Human-Centered AI (HCAI)	An AI framework with abilities (1) to explain the system's decisions or prediction to humans and (2) to support human's ability to interpret the system's explanation.	1.1 1.4
Explanation-driven interactive machine learning (XIML)	A framework with iterative user feedback on *automatic* ML algorithm solutions *to correct errors* and *align ML decisions* with users' feedback on *explanations* until reaching an acceptable solution. It is in the *traditional ML framework* of iterative applying of automatic ML algorithms. It misses the interactive opportunity to avoid errors during the training. It corrects errors only externally before and after automatic training and should be expanded by using automatic and interactive ML algorithms.	1.4
Cause of limitation of current XILM	A human needs to see multidimensional data to interact with them for constructing a model. Common lossy visualization methods like Principal Component Analysis, Multidimensional Scaling, t-SNE do not preserve all multidimensional information. It makes it practically impossible to discover full ML models interactively in these abridged data. Visual Knowledge Discovery in multidimensional spaces resolves this by using lossless General Line Coordinates, which visualize each n-D point as a graph.	1.4
Explanatory interactive machine learning (EIML)	A framework to change and improve by way of interactive explanation at *any stage* of machine learning (before, during and after training). The current implementations are limited to stages before and after training, which is conducted fully automatically by ML algorithms.	1.4
Visualization	A visual representation in 2-D/3-D *available/existing* entities (data, models, algorithms), e.g., *lossy* or *lossless* visualization of n-D training data.	1.2
Visual knowledge discovery (VKD)	A human-in-the-loop interactive AI/ML paradigm for discovery of AI/ML models in *lossless n-D data visualization space* coordinated with traditional AI/ML modeling.	1.1 1.2
Convincing ML model	A model that is *accepted* by the domain expert. Ideally it includes a human *mental/domain model* mapped to the given *ML model* in a *justified* way.	1.1
Descriptive explanation	Explanation that *only* describes the model's inner workings (a way of producing the output).	1.1
Trustable explanation	Explanation *understandable*/*comprehensible* for domain experts and that can be *potentially trusted* by the domain expert but *not accepted* by the domain expert yet.	2.1
Real/**trusted**/normative explanation	Explanation that *accurately* describes the model's inner workings and *accepted* by the user (a trustable explantion that is accepted by a domain expert).	1.1
Quasi-explanation	Explanation that uses *"foreign"* assumptions and/or terms for the domain experts. It can be inexact and unacceptable for the user. LIME and SHAP algorithms are not free from these deficiencies.	1.1 2.1
Additive quasi-explanation	Explanation that uses "foreign" assumptions for the domain experts of additivity of importance of the attributes for model explanation, e.g., SHAP algorithms can produce it.	2.2
Linear quasi-explanation	Explanation that uses "foreign" assumptions for the domain experts of linearity of importance of the attributes for model explanation, e.g., LIME algorithms can produce it.	1.1 2.1.
Overfitting model	Model that generalizes data too tight that is risky for high-stakes tasks.	1.1
Underfitting (**overgeneralized**) model	Model that generalizes data too much relative to available data that is risky for high-stakes tasks.	1.1 2.4
Lossless high-dimensional (n-D) data visualization	Visualization of n-D data that preserves all n-D information allowing a *full restoration* of each n-D point from its visualization. Commonly this visualization is conducted by mapping each n-D point to a directed graph in 2-D/3-D visualization space.	2.4 2.5 3.3
General Line Coordinates (**GLC**)	Generalization of orthogonal 2-D/3-D Cartesian coordinates for lossless visualization of n-D data, allowing coordinates to be *located* in 2-D/3-D space *anywhere* and *in any direction*. GLCs can be parallel, non-parallel, have a common origin or multiple origins, can overlap, be located in a single line, be shifted relative to each other, be straight lines of curves, can form open or closed shapes.	1.1 5.3
Heterogeneous n-D data	Adding 3 kilograms and 5 meters has no meaning and explanation in physics, as well as weighted sums like 0.2*3kg + 0.6*5m. In contrast, ML linear models freely use such weighted sums of heterogeneous data in linear regression, linear discrimination functions, SVM, shallow and deep Neural Networks, Principal Component Analysis, and in ML explainability methods like LIME. The *abstract mathematical symbols* x_1 and x_2 *hide* data heterogeneity and complicate ML explainability.	2.3
Computational ML models (**CML**)	ML models built by *computational* ML algorithms.	3.2
Visual ML models **(VML)**	ML models built by using *visual graphical tools*.	3.2

Table 4. Outline of specific concepts with core statements.

Concept	Description and core statements	Section
Trustable explanation with Bended Coordinates Decision Tree (**BC-DT**)	A trustable explanation by interactive visualization of a decision tree. It represents all actual values of attributes, for all cases, losslessly allowing to explore the confidence/certainty in the DT decision, and to improve the tree interactively.	2.5.2
Trustable explanation with Shifted Paired Coordinates – Decision Tree (**SPC-DT**)	SPC-DT has all advantages of BC-DT, in addition, it allows a user to interact with *relations* between pairs of attributes for *individual cases* in the DT within a detailed *data flow* in the DT. To the best of our knowledge, none of the DT visualization techniques now in use provide all these capabilities.	2.5.3
Shifted Paired Coordinates (**SPC**).	One of the GLC categories. In SPC, each n-D point $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is split in pairs (x_1, x_2), …, (x_{n-1}, x_n) then each pair is put to a respective Cartesian coordinate forming a point, and these points are sequentially connected forming a polyline (directed graph), which **losslessly** represents n-D point **x**.	1.2 2.5.3
GLC-Linear (**GLC-L**)	One of the GLC categories. In GLC-L, each n-D point $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is represented as a polyline (directed graph) of the segments with lengths $x_1, x_2, ..., x_n$ located under respective angles defined by coefficients of a linear discriminant function. It **losslessly** represents n-D point **x**.	3.2
Constrained LDF	Linear discriminant function (LDF) that is constrained to become interpretable.	2.3
"Foreign" distance/kernel assumption	Some automatic ML algorithms assume formulae for distances and kernels that are "foreign" for the domain and domain experts. VKD allows domain experts to observe n-D points losslessly and influence the setting of correct distances and kernels avoiding "**blind**" their assignment.	2.4
Worst-case scenarios	Most unfavorable extreme conditions. Safety-critical applications urgently need methods which work *correctly* in a worst-case scenario including adversarial attacks, difficult and unexpected inputs.	1.1 3.4
Worst-Case Linear (**GLC-WCL**) algorithm	The algorithm that finds the *lower* and *upper bounds* of the worst-case validation split by using the projections from GLC-L to locate the leftmost and rightmost misclassified n-D point.	3.3
Worst-case estimates with Shannon functions	Mathematical formulation of the worst-case estimates.	3.4
Justifiable Linear Model (**JLM**) algorithm	The ML algorithm that produces a justifiable linear model by analyzing presence of compensation property between values of attributes of n-D points.	2.3

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 from (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 discuss. 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 finally. 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 explanation is trustable** if it is understandable/comprehensible by the domain experts and can be potentially trusted by them, but not accepted by them yet. Finally, the **real/trusted/normative explanation** should be accepted by the domain experts for the deployment and use.

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 5 summarizes the key current topics in explainable AI/ML tasks, which we have divided into three categories: methodology, specific methods, and applications.

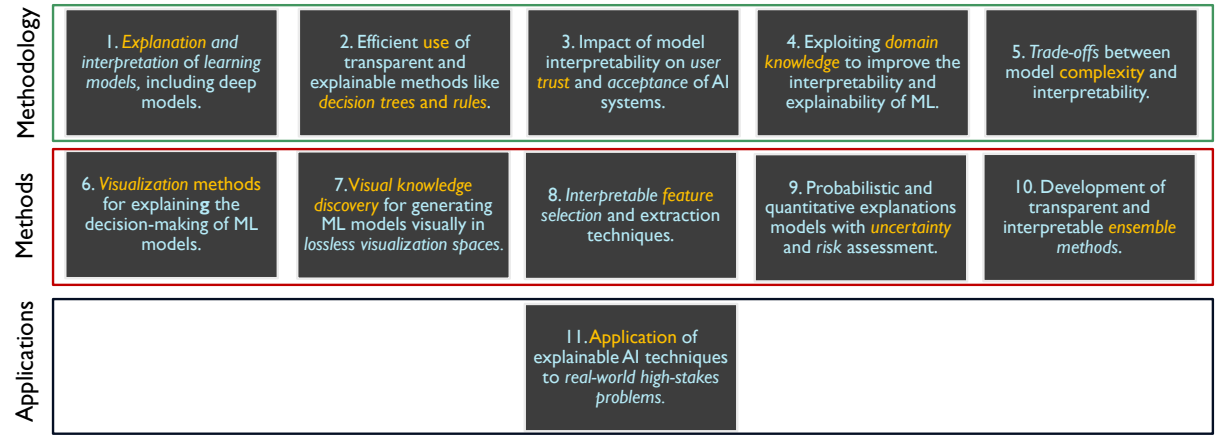


Figure 5. Topics in explainable AI/ML tasks.

The first topic in Figure 6 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* models, such as Support Vector Machine (SVM), Random Forest (RF), some linear models, and Convolutional Neural Networks (CNN).

Figure 6 illustrates these approaches.

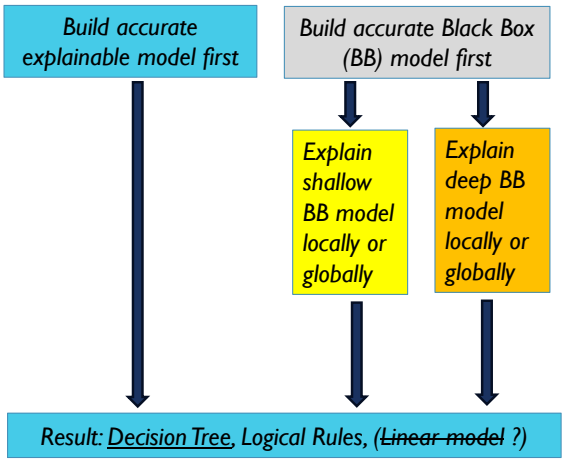


Figure 6. Explanation approached

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 6 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 and their different modifications. We show below that it is an **extreme simplification** of the actual situation with linear predictive models.

LIME (Local Interpretable Model-Agnostic Explanations) [Ribeiro, et al., 2016] is a local method intended to explain model decisions for a given case. Typically, we have an insufficient number of actual cases that are close to the given case. This case is permuted (altering it presumably slightly) and outputs of the underlying black-box ML model for these cases are computed. Next, (i) these synthetic data are used to produce a *local sparce linear model*, (ii) *alterations* are discovered that *change the decision* of the local model, and (iii) these alterations are considered as *most important* for the underlying black-box ML model for the given case. In images these alterations are visualized by a heatmap.

In this paper we discussed several deficiencies of LIME. First, the linearity of the explanation model is not well justified for ML tasks with *heterogeneous* data, second defining the *limits* of local area is extremely difficult in multidimensional space without abilities of the user to observe multidimensional data and to appropriately set up such area. Due to the important role of LIME in explainability studies we summarize claimed properties of linear LIME with our comments. Linear models are *more interpretable* than non-linear models and the reasoning of the LIME algorithm is *clear* to stakeholders, experts, and non-experts [Hashemi, 2023]. We question both these unconditional statements in this paper. Next, LIME is not well-suited for *changing boundary*. It does not specify which *features to change* or by how much. LIME does not take into account *causality* [Hashemi, 2023]. Unfortunately, these deficiencies of LIME are real.

Can we trust a model that adds 3 kilograms and 5 meters in physics or 3 molecules and 30ºC 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*30ºC 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].

It is very difficult to justify that the linear models are unconditionally intrinsically interpretable in ML. While linear models are not always, but often, are interpretable for homogeneous data, these data are not typical in machine learning problems like healthcare where interpretability is paramount. This analysis shows that it requires reexamining and **limiting the unconditional use** of popular interpretability **linear methods** and associated methods with paying more attention to **decision trees** (DT) and **logic decision rules** in propositional or first order logic (FoL) as better interpretable methods. These methods allow both (a) converting linear models to interpretable DTs and logic rules or (b) construct them from the data directly, similarly to general non-linear models discussed in section 2.

Can non-linear models resolve challenges in linear models? For non-linear decision boundaries of ML models local linear model like LIME can be too local. Can moving to non-linear local approximation resolve the issue and get (a) higher **accuracy** and (b) better prediction **explanation**? The accuracy can be reached in this way, but it is not sufficient to meet the **needs** of the explanation for **domain** experts.

This non-linearity does not link the model to the domain concepts. This issue is closely connected to the concepts of **stability** and **fidelity** of ML models based on the high **accuracy** of the models to be able to claim that the model is explainable without linking the model with domain concepts. This linking can be provided in different forms. One of the simplest ones is just asking domain experts if they understand the model explanation based on their domain knowledge. Can they give a domain meaning to the models? This is often called a usability study in many domains.

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

This section shows that popular additive explanation algorithms like SHAP produce quasi-explanations rather than real explanations if their assumptions are not tested for the application domains, which is quite typical. 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 7 shows a simple visualization with local explanations based on Tree-Explainer of a non-linear combination set of gradient boosted decision trees [Lundberg, et al., 2019].

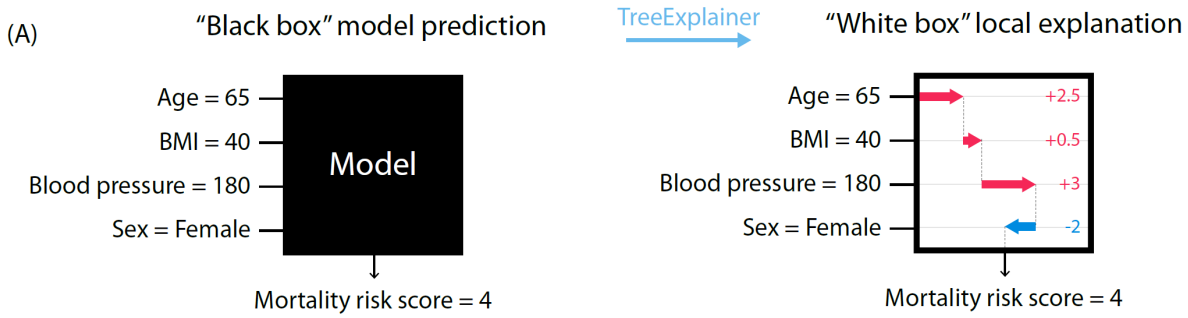


Figure 7. SHAP Tree-Explainer for tree-based models [Lundberg, et al., 2019]: linear explantion of mortality risk score 4 = 2.5*(age) + 0.5*(BMI) + 3*(blood pressure) – 2*(sex).

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 **real explanations**. Moreover, explaining a nonlinear decision tree with additive functions can oversimplify and distort the impact of nonadditive dependencies between attributes.

Figure 7 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 **additivity** 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**.

Shap uses multiple non-linear computations, but the output result is an additive function as an example in this section shows: mortality risk score 4 = 2.5*(age) + 0.5*(BMI) + 3*(blood pressure) – 2*(sex). The SHAP acronym means SHapley Additive exPlanations [Lundberg, et al., 2019]. It is additive relative to contribution $F(x_i)$ of individual features x_i, which is the goal of the SHAP, $F(\mathbf{x}) = F(x_1) + F(x_2) + \ldots + F(x_n)$, where **x** is n-D point (case). The function $F(\mathbf{x})$ is linear relative to $F(x_i)$ with all coefficients equal to 1, but functions $F(x_i)$ are not required to be linear relative to x_i. The key SHAP assumption, which causes its deficiencies, is the assumption of *additivity* (*simple linearity*) of $F(\mathbf{x})$ relative to all $F(x_i)$ as we discussed above.

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. This is the second indication that this example is likely only a **quasi-explanation**. A deeper approach is needed that would capture such **causal relations** for explantion.

Another source of a weak quasi-explanation in this example is the mortality risk score (MRS) itself. This is assigned to an individual patient, but it is not physically measured like BMI or blood pressure for this patient, but rather is *computed* from measurable attributes as a weighted combination of heterogeneous attributes. In general, the use of score concepts like MRS in the medical domain is caused by the lack of better measured characteristics. This is the third indication that this example is likely only a **quasi-explanation**. This example show that the nature of the **target**

attribute (MRS here) can create additional challenges to interpret the ML model, because it is *not directly measured* for the given case.

Unfortunately, these important negative aspects of SHAP and Tree-Explainer are not highlighted in the literature, while the base SHAP paper [Lundberg, Lee, 2017] was cited over 24,000 times by August 2024. The literature emphasizes its positive aspects like consistency/monotonicity and computational benefits of Tree-Explainer [Lundberg, et al., 2020].

The benefit of Tree-Explainer 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 but not limited to them. 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 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]. While this paper correctly pointed out deficiencies of SHAP, it missed their major cause, which is the *assumption of additivity* in evaluation of feature importance. It is *borrowed from economics* where the Shapley approach originated from and is well justified, but not in machine learning with heterogeneous data like in medicine (see above for details).

Expected relative ease of explanation by models like produced by 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. Unused features can still receive attribution [Sundararajan, Najmi, 2020]. A deeper alternative is capturing target function's topology near the input point and adding statistical tests to evaluate relationships [Fryer, Strümke, 2021].

Consider an example of a **quasi-explanation** like shown in Figure 7, where a domain expert/end user is provided importance ranks of attributes. In quasi-explanation ranks are delivered *without providing assumptions* of ranking made and without asking users *if these assumptions are acceptable* in the user's domain.

In general, the ranks of importance of the attributes are specific to: (1): the *ML algorithm* used to build the predictive ML model, (2) the *algorithm that ranks attributes* and (3) *data* used to build the ML model and ranks. For instance, the SVM and Random Forest algorithms with SHAP can lead to different ranks on the same data. These specifics include **assumptions** that these algorithms made like the type of SVM kernel and SHAP additivity. Typically, a user who is not a data scientist is unaware of assumptions and cannot judge how relevant or irrelevant or "foreign" to the user's domain these assumptions and their consequences are. Often this is also challenging for data scientists too.

Making end users sophisticated data scientists is quite unrealistic. This situation requires **reexamination** of the whole endeavor of providing importance ranks of attributes to end users to explain the ML models as a **single order**. It seems that alternative explanation approaches like providing **logical rules** and decision trees that approximate black-boxes, and providing similar and dissimilar cases are more appropriate for these end users to get **real explanations not quasi-explanations**.

Consider a branch of the decision tree starting from its root with attribute x_1, then attribute x_2 in the second level and attribute x_3 in the third level in that branch. This branch classifies 3-D case $\mathbf{x} = (x_1, x_2, x_3)$ as follows:
If $x_1 < 5$ & $x_2 > 10$ & $x_3 < 7$ then **x** belongs to class A. No other of total 6 attributes are involved in this branch.
Also assume that we have the statistics:

(1) 600 cases out of total 1000 training cases satisfy property $x_1 < 5$ (60%),
(2) 500 cases out of 600 cases from (1) satisfy property $x_2 > 10$ (83.33%),

(3) 500 cases out of 500 cases from (2) satisfy property $x_3 < 7$ (100%)

This branch has 100% precision for all cases of class A. Here only attributes x_1-x_3 are informative and can get non-zero importance ranks. Which one of them should get the highest rank? This is an **incorrect question** here, as we show below, while it is typically asked and "answered" by many ranking algorithms. The attribute x_1 is most important at the level 1. The attribute x_2 is most important at the level 2 and the attribute x_3 is most important at the level 3 at this tree branch. Assume that this is a result of the decision tree construction using Gini Index or entropy. Can we combine these three different answers to a single answer by ordering attributes x_1-x_3, say, by stating that x_1 is more important than x_2, and x_2 is more important than x_3? We can derive it by using the number of cases covered by clauses with each of them (600, 500, 500), but clauses with x_2 and x_3 cover the equal number of cases and we should keep them equally important. In this ordering we ignored the important level information.

Presented orderings of attributes based on simple statistics can be considered as **trustable explanations** because the simple interpretable statistics are used to build it, and users can understand it without need to become data scientists. Next, a user may select only a subset of attributes as most important because the other ones are assumed already fixed for a range of cases which user is considering.

In the examples above we oversimplified the explantion situation with selecting a **single ranking of attributes** (one-dimensional ranking) instead of **multiple rankings** with multiple criteria. When we ask a user to confirm ranking from the user's domain perspective without telling a user the single criterion we used to order attributes, we ask an *incorrect question* that cannot be answered by yes/no. However, the popularity of the **incorrect and oversimplified approach** with a single ranking of importance of attributes is caused exactly by its **simplicity to apply** by available software.

Our argument is that there is no reason to use this oversimplified approach with the single ranking of attributes without providing an interpretable for the user ranking criterion. Branches of decision trees and logical rules **do not require** such ordering at all to become **real explanations** of the black-box models, which they approximate. Domain experts understand the clauses in them as far as attributes are interpretable in the application domain. More such explanations using decision trees can be found in section 2.5.

Now consider not a branch of the decision tree but a logical rule: If $x_1 < 5$ & $x_2 > 10$ & $x_3 < 7$ then **x** belongs to class A, where clauses are computed independently with the following statistics:

(1) 600 cases out of total 1000 training cases satisfy property $x_1 < 5$ (60%),
(2) 700 cases out of total 1000 training cases satisfy property $x_2 > 10$ (70%),
(3) 600 cases out of total 1000 training cases satisfy property $x_3 < 7$ (60%)
(4) 500 cases out of total 1000 training cases satisfy all three clause $x_1 < 5$ & $x_2 > 10$ & $x_3 < 7$ (50%)

This rule has 100% precision to cases of class A as the branch of the decision tree above.

The statistics of this rule differ from statistics of the tree branch above. Based on the number of cases covered by clauses the order would be x_2, then x_1 and x_3 as equally important. An alternative ordering can be based on how closely each clause covers 500 actual cases of the class A. In this situation x_1 and x_3 are closer to 500 cases covering 600 cases each. So, they will be considered as the most important and x_2 as a less important attribute. Again, depending on the criteria used to order attributes we get different orders of importance.

How to find out if an explanation is a real explanation instead of a quasi-explanation? A **quasi-explanation** includes terms and/or assumptions that are *not present* in the domain of the ML task at hand ("**foreign**" **terms and/or assumptions**), which are *difficult or impossible to justify in the user's domain*. The **real explanations do not** include such "foreign" terms and/or assumptions. Next, the real explanation must be accepted by the domain expert/end user.

An example of "foreign" assumptions is that **all points** of the feature space correspond to **legitimate cases** in the user's domain. For many domains it is a severe overgeneralization of real data [Kovalerchuk, Grishin, 2018]. Other examples are terms like hidden layers, kernels, ReLU (rectified linear unit) activation function, SHAP assumption of additivity, and LIME assumption of linearity.

2.3. Interpretable linear models

The purpose of a Justifiable Linear Model approach is to make current popular linear explanations really justifiable as the name suggests. It allows to convert linear **quasi-explanations** to linear explanations *confirmed* by a domain expert and available data to become a **real explanation**. This is one of the main points of this paper.

Below we show how interpretable linear ML models can be built. We define the criteria when the linear model can be justified, and when it cannot be justified, therefore requiring modification. 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**. Here **u** and **w** are **compensation points** for each other. The 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 the training data contain points **u**, but do not contain compensation 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 domain experts have **difficulties** to understand and interpret linear model in this situation, where the experts' domain practice does not contain compensation 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 outlined below:

> Step 1: For each case like **u** = (2.5) *search* for similar cases, like **z** = (2.3, 4.9) and for the compensation cases like **w** = (5, 2).
> Step 2: If the number of compensation cases **w** is *comparable* with the number of **u** or **z** cases, then we build a standard LDF on these data.
> Step 3: If the number of compensation cases **w** is *much smaller* than the number of **u** or **z** cases, then we *find constraints* as explained below and build **a constrained** LDF (model **CLDF)** on these data.
> 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), e.g., by algorithms in [Huber, et al., 2024]. This makes these linear models more easily interpretable for the domain experts.

Step 3 allows for putting constraints on attribute values of cases, to avoid the model overgeneralization to **w** cases. It can be conducted by adapting the hyperblock approach [Huber, et al., 2024; Hayes, et al., 2024] with controlling presence of the compensation cases. Below is an example to illustrate step 3. For a case **u** we check the presence of compensation point **w**. In the absence of such **w**, we limit values of attributes x_1 and x_2, by vicinities of case **u**, e.g., $x_1 \in [1, 3]$ and $x_2 \in [4, 6]$. These intervals include **u** = (2, 5) but do not include **w** = (5, 2). Respectively, the CLDF model can 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 with all areas, 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 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} and L_{i2} with $x_i \in [L_{i1}, L_{i2}]$. The CNDF can use an iterative approach to find such hyperblocks. This creates a **constrained LDF (**CLDF) in a general form:

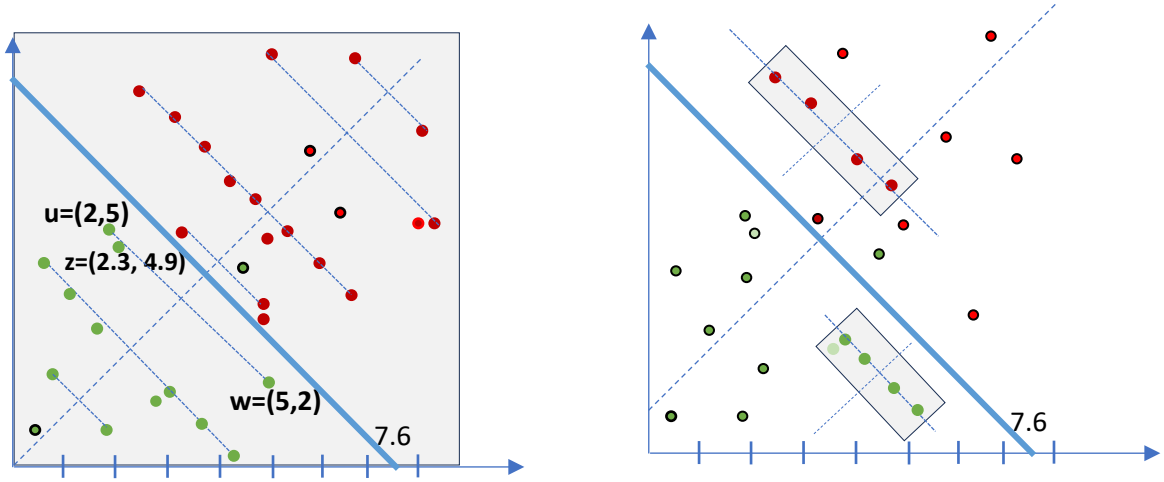
$$\text{If } f(\mathbf{x}) > T \ \&\ (\mathbf{x} \in HB_1 \text{ or } \mathbf{x} \in HB_2 \ \ldots\ \mathbf{x} \in HB_k) \text{ then } \mathbf{x} \text{ belongs to class 1.} \quad (3)$$

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 $\mathbf{x}$ 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. It can be performed by the same algorithm that builds hyper-rectangles (hyperblocks) in step 3 from [Huber, et al., 2024].

Figure 8 illustrates the JLM algorithm showing the situation (a) where global linear explanations are justifiable and situation (b) where it is not justifiable. In Figure 8 the points that do not have compensation points have a black outline. On the left picture, *almost every* point has a compensation point, and some points have *similar* points, e.g., similar points **u** and **z** have a compensation point **w**. It shows that attributes x_1 and x_2 are mostly *mutually exchangeable*. In contrast, on the right picture, attributes x_1 and x_2 are not globally mutually exchangeable. Here only points in the grey areas have local compensation points.

Respectively, situation (a) allows building a **standard LDF** to get an interpretable LDF model, but situation (b) requires building a **constrained LDF** to get an interpretable LDF model. Respectively, Figure 8 shows a linear discrimination function with a blue discrimination line: x + y = 7.6, which separates red and green classes. This traditional linear discrimination function is justified for the situation in Figure 8a because most of the points there have compensation points. In contrast in Figure 8b, points do not have such global compensation points and this LDF is not justified.

This figure shows that for a dataset to be qualified for a global linear classifier we need points located in lines parallel to the discrimination line in both classes for most of the points. These lines are exposed by grey strips. A dataset that satisfies a Gaussian distribution for each class, can satisfy the compensation property due to the symmetry of the Gaussian distribution. The points far away from the centers of these distributions can be rejected due to their low probabilities to control overgeneralization.



a) Globally mutually exchangeable attributes x_1 and x_2 in grey area where almost every point has a compensated point relative to the global diagonal dotted line, e.g., points **u** and **z** have a compensated point **w**.

(b) Locally mutually exchangeable attributes x_1 and x_2 in the local grey areas. Only points in these areas have local compensated points.

Figure 8. Situations where global linear explanations are justifiable (a) and not justifiable (b).

The concept of compensation points has useful applications beyond this algorithm. It allows expanding a set of training data without explicitly asking a domain expert to label more individual cases, which is a slow and labor-intensive process. Instead, we ask a domain expert to *confirm a compensation property* in the specific area, or for all feature space relative to some base characteristics. It allows to expand training data *automatically* by adding compensation cases for all training cases in the areas where compensation was confirmed by the domain expert.

The compensation property also allows for an alternative view of *similar cases* as *local compensation cases.* It can help to define close cases more meaningfully. Next, the compensation can be non-linear and symmetric in several ways and can be attribute or case specific. For example, for pair (140, 80), pair (120, 100) might be a compensation

pair, but (160, 60) may not, if in the first pairs 120 and 80 are assumed to be "good" values, but in the third pairs both values are not "good" for the user.

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. They concluded 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 a classical fuzzy logic problem of **defining a linguistic variable** with values like, "far away", "nearby", and, 'intermediate". Note that, here we have a more complex multidimensional linguistic variable, not a one-dimensional linguistic variable, for individual attributes.

While often a domain expert can judge which points are nearby in 2-D space, it is difficult in a typical **multidimensional machine learning space** because the expert cannot see n-D cases. We have only the numerical values of distances and considering them as "far away" or "nearby" is quite subjective. The same is applicable to distance used in clustering approaches. Thus, in the high dimensional space we are fundamentally, "blind" 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. We will illustrate it later in section 3.1 in detail.

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 considerations above show 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 to provide **trustable** and **real explanations, not quasi-explanations**, which we presented in section 2.2 with examples contrasting quasi-explanations and real explanations. 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 next topic is efficient use

of transparent and explainable methods like decision trees (DTs) and rules. This has the following challenges to get real explanation with DTs and rules:

(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*?

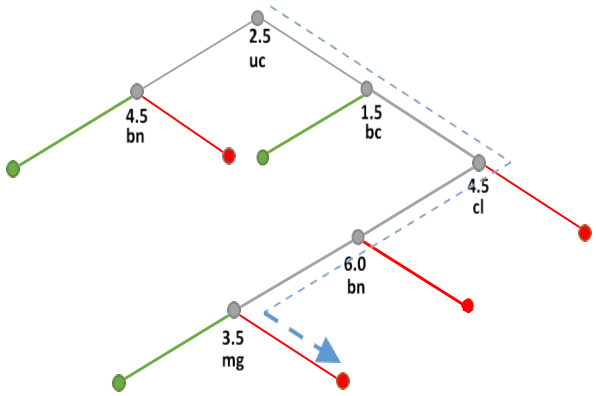
The known approaches to address these challenges are computational and interactive with extensive use of visualizations [Gupta et al., 2022; Kovalerchuk, Dunn, et al., 2024]. The challenge (1) has been extensively addressed in the ML literature for decades with multiple algorithms available and new developments 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.

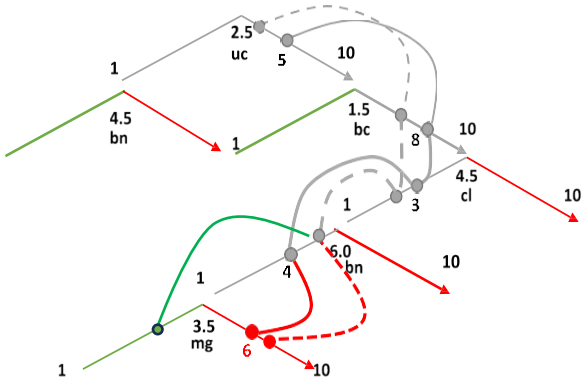
2.5.2. Bended Coordinates Decision Tree (BC-DT) method

While all decision trees conceptually can provide **real explanation** for ML, the level of domain expert involvement in DT discovering and analyzing is very different with different visualization as **examples** in Figures 9-11 show. Traditional visualizations allow much less involvement of domain experts than newer methods presented below.

Figure 9(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 9a, 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.



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



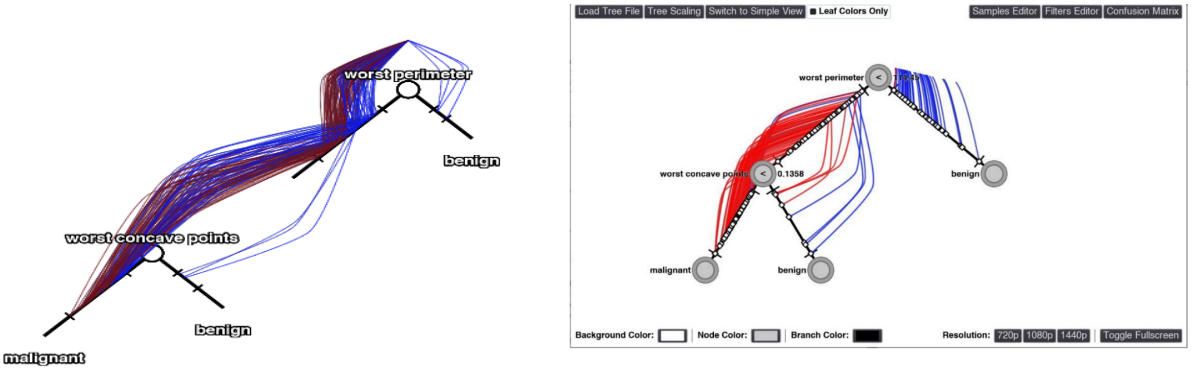
(b) DT with edges as Bended Coordinates in disproportional scales. The curved lines are cases that reach the DT malignant edge with different certainties due to the different distances from the threshold node.

Figure 9. Explanation with Bended Coordinate Decision Tree (BC-DT) [Kovalerchuk, 2023].

In contrast Figure 9(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. Here, the dotted case is a new case that needs to be predicted, and a non-dotted case is available 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).

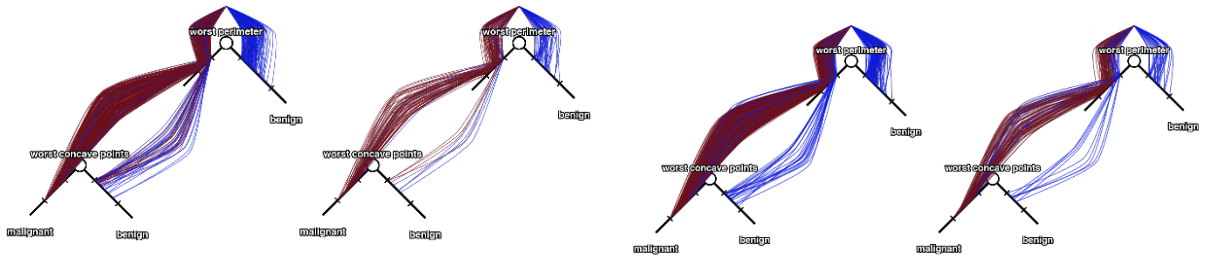
To address (8), which is supporting interactive modification of the DT and rules by the domain expert to get desired accuracy and confidence, the BC-DT method allows a user interactively *adjust thresholds*. To address (9), which is supporting analysis that training data and validation data are from 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. 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 9b, 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 9b these intervals are made equal by different (disproportional) scaling them to keep this visualization like a traditional visualization on Figure 9a. In the proportional scaling, the sides can be of different lengths if the threshold point is not in the middle of the interval. Figure 10 shows DT with cancer data for both options implemented.



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

Figure 10. BC-DT with proportional edges (a) and disproportional edges (b) serving as Bended Coordinates [Kovalerchuk, Dunn, et al., 2024].

Figure 11 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 11. 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 to get a **trustable** and **real explanation** 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.

It uses the **Shifted Paired Coordinates** (**SPC**) representation described in section 1.2 as one of the categories of General Line Coordinates (GLC) for 4-D data and illustrated in Figure 3 there. In general, 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 12 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. 12, 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.

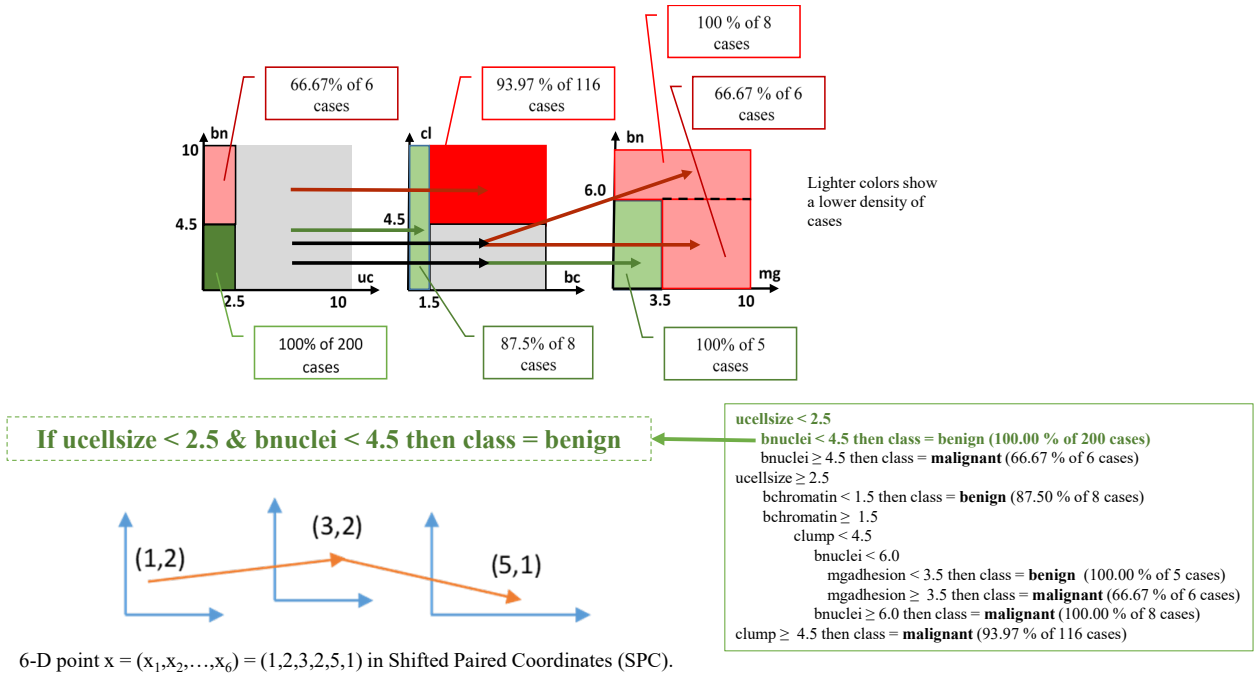


Figure 12. Shifted Paired Coordinates – Decision Tree (SPC-DT) method.

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 12 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 12 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).

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 13 illustrates these capabilities for the cancer dataset to produce a user-friendly DT.

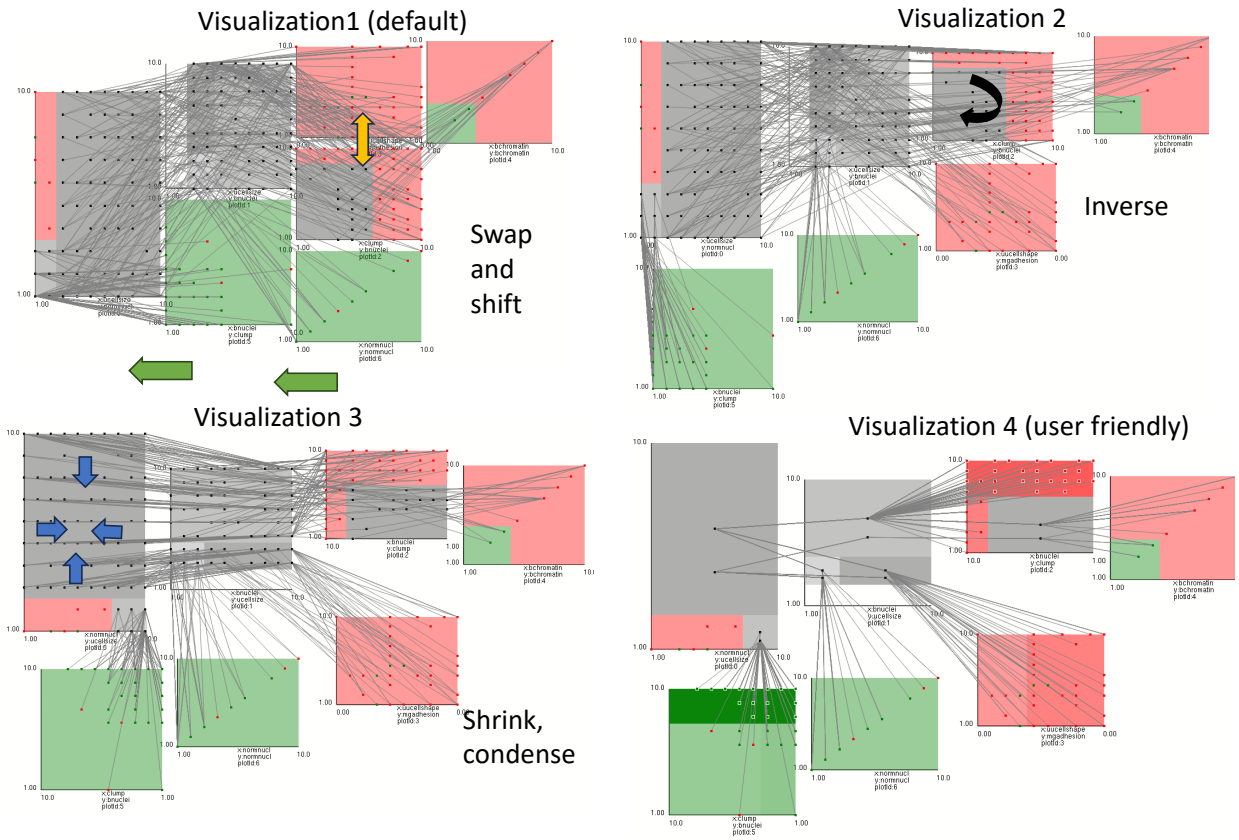


Figure 13. Illustration of interactive SPC-DT method for cancer data set to produce a user-friendly DT.

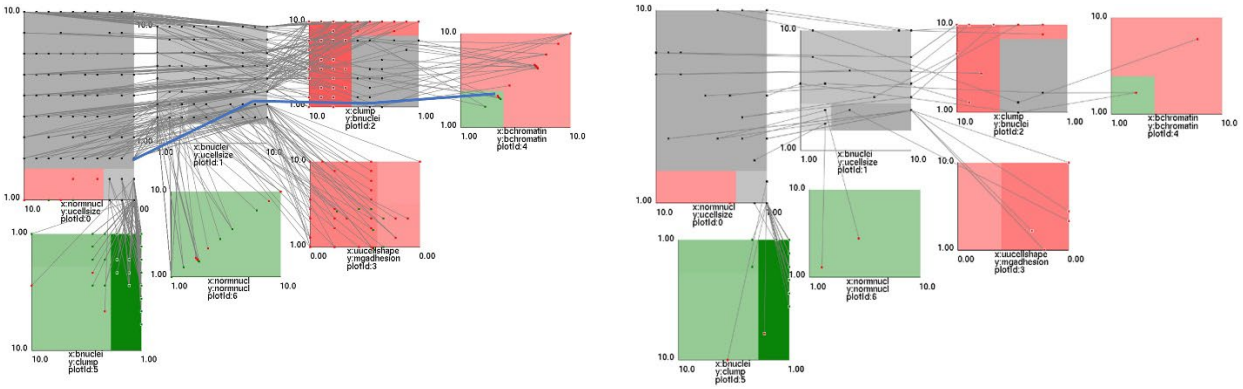
The gray areas on each pair of coordinates are areas where a case's class is unknown yet at the current DT level. To depict the various outcomes of cases in gray areas, several grayscale shades are used. The DT has classified some cases as, "malignant", while other cases have been classified as, "benign", 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 14 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. 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.



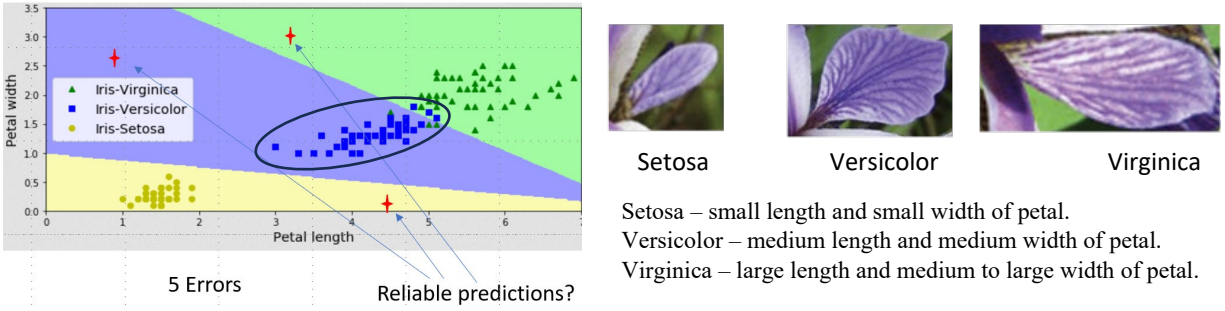
(a) Training data (90%) in SPC-DT. (b) Testing data (10%) in SPC-DT.

Figure 14. Comparison of training and testing cancer data in SPC-DT with 90/10 split of all data to training and testing data.

3. Avoiding Unreliable Predictions for High- Stakes Tasks

3.1. Overgeneralization as a challenge for high-stakes task

Data overgeneralization has multiple negative consequences for high-stakes tasks. This overgeneralization is one of the sources of unreliable predictions with exaggerated accuracy [Gao, et al., 2019]. Below we discuss these challenges and ways to overcome them. Figure 15a illustrates data overgeneralization by linear Logistic Regression (LR) models.



(a) Logistic regression classification of Iris [Big data, 2018]. (b) visual and verbal Setosa, Versicolor and Virginica Iris petals.

Figure 15. Human visual and verbal Generalization vs. Machine Learning generalization: Iris dataset example.

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 15. 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 are another source of danger of overgeneralization for high-stakes tasks.

It is a deficiency of LR and many other **ML algorithms** like Random Forest, Naïve Bayes, and Support Vector Machine (SVM), which **classify 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 15b, 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 15a is not unique, many software ML packages, like MLxtend [Raschka, 2023] produce, "decision regions", without any warning as to how severe such, "decision regions", overgeneralize. Further 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. *Computational ML models* **(CML)** 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 Figure 15b as a basis for the learning constraints 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 16 illustrates the visual LRO approach. It uses the SPC-DT model visualization already described above and illustrated for cancer data in Figures 13 and 14. Now this is applied to the Iris dataset presented in Figure 15 with the difference from Figure 15 being Figure 16 is showing more than only two attributes. Figure 16 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 16 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].

The performance of ML classical algorithms is commonly evaluated by accuracy of the prediction or by error rates, which are forms of the **performance loss functions/cost functions** relative to the ideal accurate prediction. This measure of performance is generalized to include learning with reject, where the loss not only assigned to the wrong predictions but also to the rejection of prediction. For instance, we can have a situation where a case that can be predicted correctly was refused to be predicted. Respectively, it is possible to compare the loss function of the

classifiers without rejection or with different rejection algorithms and to explore when rejection will provide less loss function like it is done in [Cortes, et al., 2016]. This and other studies were able to demonstrate the lower loss with rejection than without it, showing efficiency of learning with rejection.

Some loss functions are based on the magnitude of the value of the classifier target function focusing on rejection of cases with low probability of correct classification. Alternative loss function can add other characteristics. One of the promising characteristics is the ratio of the size of the training data and the size of the used feature space. Rejection allows to increase this ratio relative to standard ML models produced without any rejection. It is done by limiting the whole feature space to a subspace. For instance, it is shown in section 2.4. that the given Iris data cases represent only 0.026% of the total number of points in its 4-D discrete space. Learning with reject allowed dramatically improve this ratio. Figure 16 highlights the areas of the total space used for actual prediction with dotted lines. In general, the design and selection the loss function is a research issue, where a combination of computational and visualization approaches is promising as this section shows.

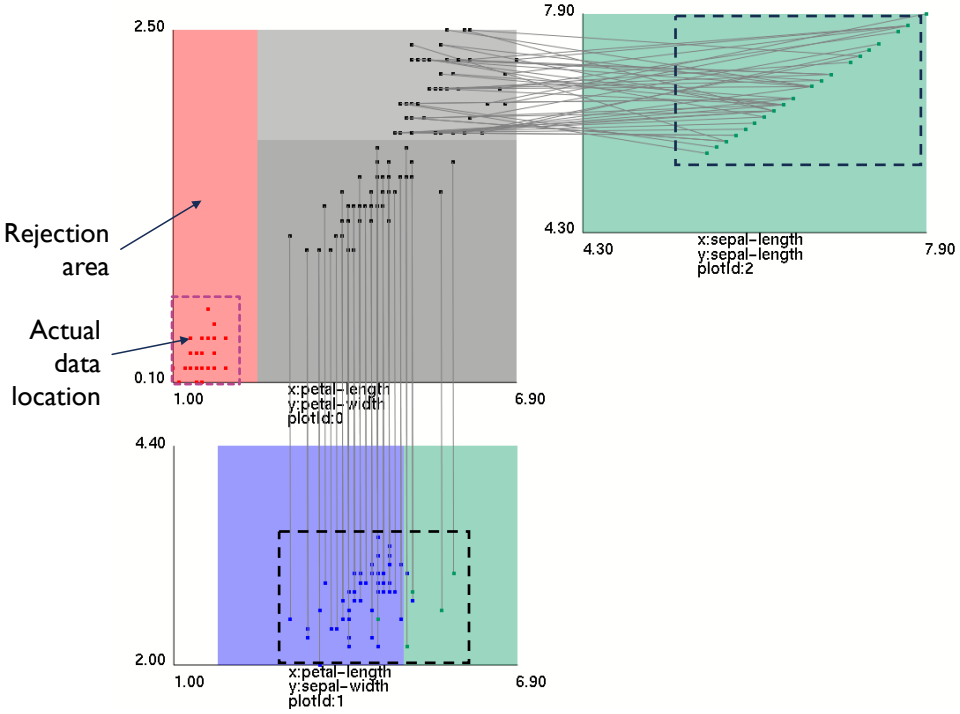


Figure 16. Learning with Reject Option (LRO) decision tree ML model in SPC-DT on Iris dataset.

Figures 17 and 18a provide another example of learning with rejections option. This is a visual method based on General Line Coordinates-Linear (GLC-L) [Kovalerchuk, 2018]. Figure 17a 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 17b is moving the 2nd vector $\mathbf{x}_2$ to the top of the vector $\mathbf{x}_1$, 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 17c 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 17d).

Figure 18a 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.

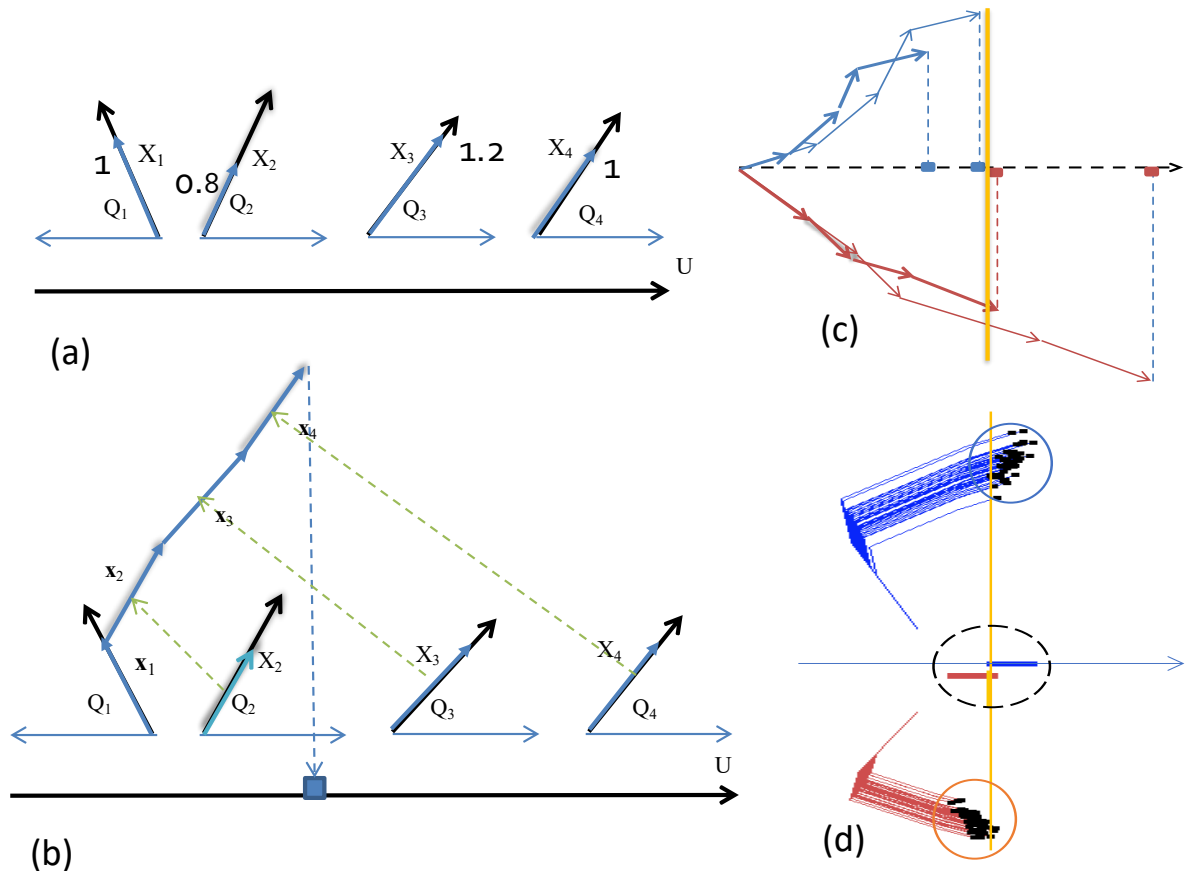
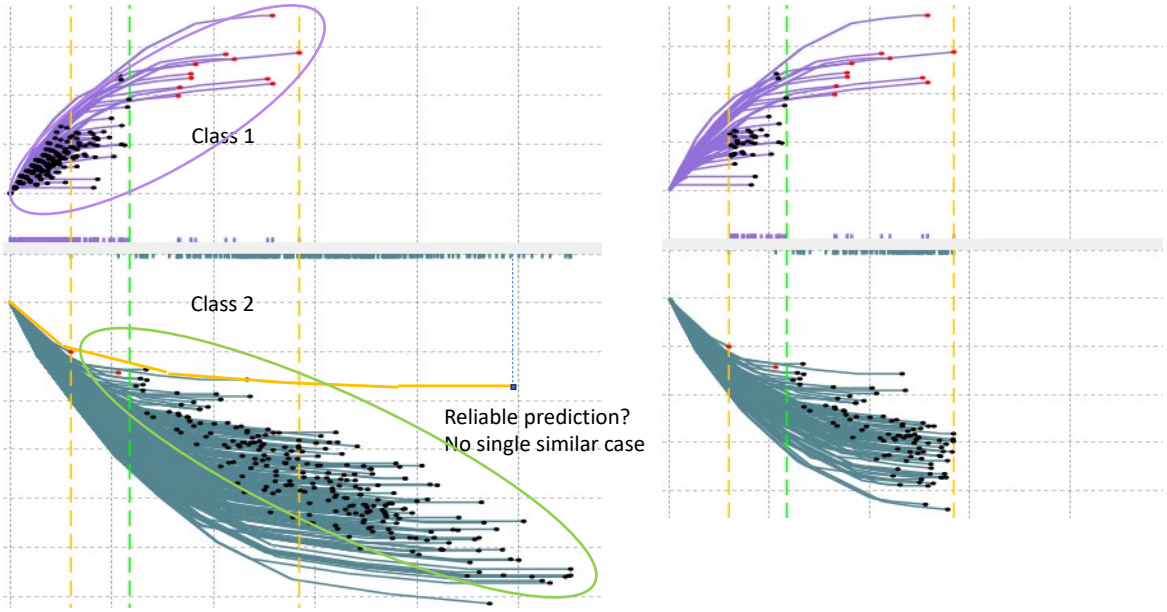


Figure 17. Learning with rejection option in General Line Coordinates-Linear (GLC-L).



(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 18. 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

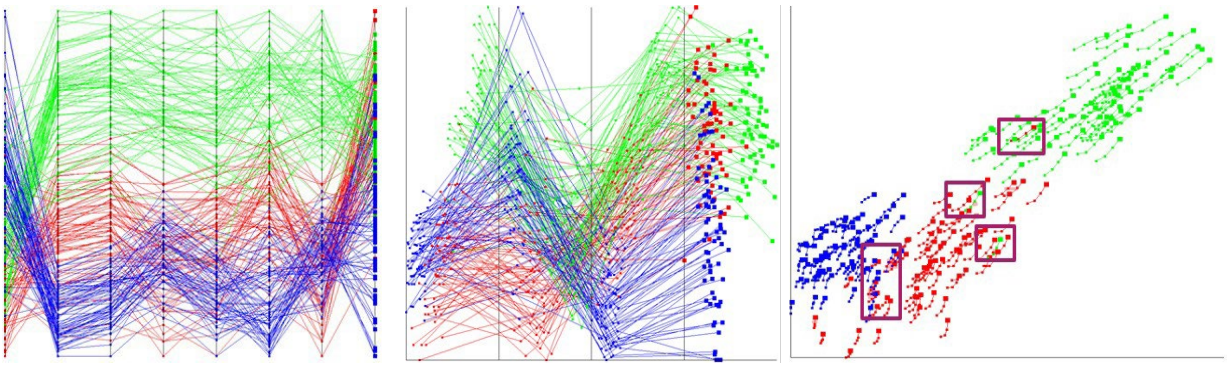
The key idea of the **visual worst-case approach** is (1) selecting interactively in lossless n-D data visualization **most difficult n-D cases** and then (2) computing the **accuracy** of different algorithms on these most difficult data when the algorithm is trained on the remaining data. We search 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 18b.

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 19 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 19c. It was done with scaling of the attributes.



(a) Seeds on Parallel Coordinates (PC).

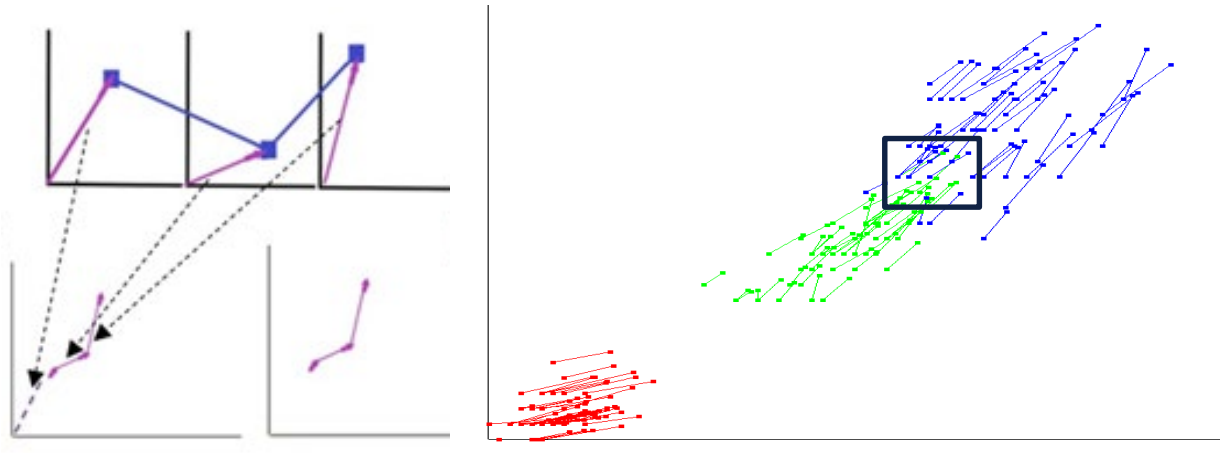
(b) Seeds on Shifted Paired Coordinates (SPC).

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

Figure 19. 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 20. This is similar to GLC-L as illustrated in figures 17 and 18. 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 20a 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 20a), (4) remove the first magenta line as not necessary to have a losses representation of the n-D point in DSC2, and optional (5) scale some pairs of coordinates allowing to **visual granular patterns** to be more visible. Figure 20b 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 20. Dynamic Scaffold Coordinates based on Shifted Paired Coordinates (DSC2).

Table 5 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 19c 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).

Similar results have been obtained for Wisconsin Breast cancer (WBC) data. Table 6 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 57.6% accuracy.

Table 5. Worst-case split for Seeds vs. 10-fold cross validation obtained with DSC-2 visualization.

Model	**10-fold Average Accuracy (%)**	**Worst-case Accuracy (%)**
Decision Tree (DT)	91.0	57.1
Support Vector Machine (SVM)	**97.1**	38.1
Random Forest (RF)	91.9	**61.9**
k-Nearest Neighbors (kNN)	88.6	23.8
Logistic Regression (LR)	91.4	38.1
Naïve Bayes (NB)	89.0	57.1
Multilayer perceptron (MLP)	89.5	23.8

Table 6. Worst-case split for WBC vs. 10-fold cross validation obtained with DSC-2 visualization.

Model	**10-fold Average Accuracy (%)**	**Worst-case Accuracy (%)**
Decision Tree (DT)	94/7	62.1
Support Vector Machine (SVM)	97.1	62.1
Random Forest (RF)	96.6	65.2
k-Nearest Neighbors (kNN)	**97.2**	60.6
Logistic Regression (LR)	96.8	63.6
Naïve Bayes (NB)	96.1	**72,7**
Stochastic gradient descent (SGD)	96.2	57.6
Multilayer perceptron (MLP)	95.5	60.6

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,

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

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, 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 is expanding the presented approaches with a wider set of analytical and Visual Knowledge Discovery methods in combination with multiple emerging machine learning methods. Major current and future research topics in interpretable AI/ML models for high-stakes tasks are listed and illustrated in section 1.1. New extensive studies are needed for high-stake tasks in *Methodology* of explaining black-box models, *Computational methods* to explain AI/ML models, *Human-in-the-loop,* and *Visual Knowledge Discovery*. Visual Knowledge Discovery 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. The major future topics in VKD are presented in section 1.2. The spectrum of machine learning models that are widely accepted as interpretable need to be expanded beyond decision trees and rule-based systems to systems based on lossless visualizations of high-dimensional data. Expansion of complex large high-stakes ML tasks that can solved by interpretable methods is a ML grand challenge. A combination of traditional computational ML method and Visual Knowledge Discovery (VKD) is a promising approach for the further research in this area.

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. Adebayo J, Gilmer J, Muelly M, Goodfellow I, Hardt M, Kim B. Sanity Checks for Saliency Maps in Advances in Neural Information Processing Systems 31, 9505-9515, 2018.
2. 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.
3. Bartlett PL, Wegkamp MH. Classification with a Reject Option using a Hinge Loss. Journal of Machine Learning Research. 2008, 1;9(8).
4. Big Data and Machine Learning, 2018, http://www.cnblogs.com/luweiseu/p/7826679.html
5. Capel T, Brereton M. What is human-centered about human-centered AI? A map of the research landscape. In: Proceedings of the 2023 CHI conference on human factors in computing systems 2023 Apr 19 (pp. 1-23).
6. Cortes C, DeSalvo G, Mohri M. Learning with rejection. InAlgorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27 2016 (pp. 67-82). Springer International Publishing.
7. Craik KJ. The nature of explanation. CUP Archive; 1967.
8. DARPA XAI program, 2016, https://www.darpa.mil/program/explainable-artificial-intelligence.
9. Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608. 2017 Feb 28.
10. Fails JA, Olsen Jr DR. Interactive machine learning. InProceedings of the 8th international conference on Intelligent user interfaces 2003 Jan 12 (pp. 39-45).
11. Finzel B. Human-Centered Explanations: Lessons Learned from Image Classification for Medical and Clinical Decision Making. KI-Künstliche Intelligenz. 2024 Feb 14:1-1.
12. Freiesleben T, Grote T. Beyond generalization: a theory of robustness in machine learning. Synthese, 2023 Sep 27;202(4):109.
13. Fryer D, Strümke I, Nguyen H. Shapley values for feature selection: The good, the bad, and the axioms. IEEE Access. 2021 ;9:144352-60.
14. 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).
15. 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.
16. Goodfellow, I. J., Shlens, J., & Szegedy, C. Explaining and harnessing adversarial examples. 2014, arXiv preprint arXiv:1412.6572.
17. Gori M, Betti A, Melacci S. Machine Learning: A constraint-based approach. Elsevier; 2023 Mar 1.

18. Guo L, Daly EM, Alkan O, Mattetti M, Cornec O, Knijnenburg B. Building trust in interactivmachine learning via user contributed interpretable rules. InProceedings of the 27th international conference on intelligent user interfaces 2022 Mar 22 (pp. 537-548).
19. Gupta A, Saunshi N, Yu D, Lyu K, Arora S. New definitions and evaluations for saliency methods: Staying intrinsic, complete and sound. Advances in Neural Information Processing Systems. 2022 Dec 6;35:33120-33.
20. Hashemi M. Who wants what and how: a Mapping Function for Explainable Artificial Intelligence, 2023, ttps://arxiv.org/abs/2302.03180v1
21. 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.
22. He X, Zhao K, Chu X. AutoML: A survey of the state-of-the-art. Knowledge-based systems. 2021 Jan 5;212:106622.
23. 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).
24. Hohman F, Head A, Caruana R, DeLine R, Drucker SM. Gamut: A design probe to understand how data scientists understand machine learning models. InProceedings of the 2019 CHI conference on human factors in computing systems 2019 May 2 (pp. 1-13).
25. 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.
26. 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.
27. Inselberg A, Dimsdale B. Parallel coordinates. Human-Machine Interactive Systems. 2009:199-233.
28. Johnson W. B. , L, Extensions of Lipschitz mappings into a Hilbert space, Contemp Math 26 (1984), 189–206
29. Kovalerchuk B., Visual Knowledge Discovery and Machine Learning, Springer Nature, 2018
30. 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.
31. 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.
32. Kovalerchuk B, Nazemi K, Andonie R, Datia N, Banissi E, (Eds). Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery, Springer, 2022
33. Kovalerchuk B, Nazemi K, Andonie R, Datia N, Bannissi E, editors. Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery. Springer Nature; 2024.
34. 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
35. 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.
36. 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.
37. 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.
38. 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.
39. Kulesza T, Burnett M, Wong WK, Stumpf S. Principles of explanatory debugging to personalize interactive machine learning. In: Proceedings of the 20th international conference on intelligent user interfaces 2015, pp. 126-137.
40. 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.
41. 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.
42. 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.
43. 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/
44. 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.
45. 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.
46. 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
47. 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.
48. Molnar. C. 2020. Interpretable Machine Learning. https://christophm.github.io/interpretable-ml-book/.
49. 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.
50. Michel P, Ruder S, Yogatama D. Balancing average and worst-case accuracy in multitask learning. arXiv preprint arXiv:2110.05838. 2021.
51. Raschka, S. MLxtend, 2023. https://rasbt.github.io/mlxtend/
52. Ribeiro MT, Singh S, Guestrin C. " Why should I trust you?" Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining 2016, pp. 1135-1144.
53. 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
54. 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.
55. 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.
56. Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature machine intelligence. 2019 May;1(5):206-15.
57. 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.
58. Schmid U, Finzel B (2020) Mutual explanations for cooperative decision making in medicine. Künstliche Intell 34(2):227–233
59. Shneiderman B. Human-centered AI. Oxford University Press; 2022.
60. Slany E, Scheele S, Schmid U. Hybrid Explanatory Interactive Machine Learning for Medical Diagnosis. InIFIP International Conference on Artificial Intelligence Applications and Innovations, 2024 Jun 21 (pp. 105-116). Cham: Springer Nature Switzerland.
61. Sundararajan M, Najmi A. The many Shapley values for model explanation. In: International conference on machine learning 2020 Nov 21 (pp. 9269-9278). PMLR.
62. Teso S, Kersting K. Explanatory interactive machine learning. InProceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society 2019 Jan 27 (pp. 239-245).
63. Teso S, Alkan Ö, Stammer W, Daly E. Leveraging explanations in interactive machine learning: An overview. Frontiers in Artificial Intelligence. 2023 Feb 23;6:1066049.
64. Theisen RC. Beyond Worst-Case Generalization in Modern Machine Learning, Doctoral dissertation, University of California, Berkeley, 2023, https://escholarship.org/uc/item/62j5g7vd.
65. VKD, CWU Visual Knowledge Discovery Lab software, 2024, https://github.com/CWU-VKD-LAB,
66. 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
67. Wolberg W., Breast Cancer Wisconsin (Original), 1992, https://archive.ics.uci.edu/dataset/15/breast+cancer+wisconsin+original
68. 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.
69. 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).
70. Zhang J, Bargal SA, Lin Z, Brandt J, Shen X, Sclaroff S. Top-down neural attention by excitation backprop. International Journal of Computer Vision. 2018 126(10):1084-102.
71. 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.
72. Zytek A, Liu D, Vaithianathan R, Veeramachaneni K. Sibyl: Understanding and addressing the usability challenges of machine learning in high-stakes decision making. IEEE Transactions on Visualization and Computer Graphics. 2021 Sep 29;28(1):1161-71.https://doi.org/10.1109/TVCG.2021.3114864