Int. J. Appl. Math. Comput. Sci., 2026, Vol. 36, No. 3, 407–418
DOI: 10.61822/amcs-2026-0026

ALIKE PARTS: A FEATURE–INFORMED APPROACH TO LOCAL AND GLOBAL PROTOTYPE EXPLANATIONS

JACEK KAROLCZAK [a,*], JERZY STEFANOWSKI [a]

[a] Institute of Computing Science
Poznan University of Technology
ul. Piotrowo 2, 60-965 Poznań, Poland
e-mail: {jacek.karolczak,jerzy.stefanowski}@cs.put.poznan.pl

Prototype-based explanations offer an intuitive, example-based approach to support the interpretability of machine learning black box classifiers but often lack feature-level granularity. We introduce a framework that integrates feature importance at two levels to address this gap. First, for local explanations, we propose *alike parts*—a method that uses feature importance scores to highlight the most relevant, shared feature subsets between a classified instance and its nearest prototype, guiding user attention. Second, we augment the global prototype selection objective function with a feature importance term to actively promote diversity in the feature attributions of the selected prototypes. Experiments on six benchmark datasets show that this augmented selection process maintains or, in some cases, increases the prediction fidelity of the surrogate model, suggesting that feature diversity does not compromise model fidelity.

Keywords: explainable artificial intelligence, prototype-based explanation, feature importance, user attention guidance.

1. Introduction

The growing application of modern artificial intelligence (AI), particularly machine learning (ML), to high-stakes real-world problems, especially those that impact human welfare or critical systems, necessitates requirements beyond predictive accuracy. A primary concern is establishing human trust in the safe and appropriate use of these systems. This need has led to the formalization of the principles known as trustworthy AI (Kaur *et al*., 2022). Within this framework, explainability has been identified as a cornerstone requirement (Chamola *et al*., 2023). Note that many of the highest-performing ML methods are black-box models, which inherently lack transparency, failing to provide insight into the rationale behind their decisions or the internal logic driving their decisions. This fundamental conflict between the critical need for transparency and the opaque nature of leading models is the main driver for the growing field of explainable artificial intelligence (XAI).

The field of XAI is a broad and active research area focused on developing methods to explain black-box models. These approaches offer various types of explanations and are mainly categorized by their scope. *Local explanations* provide the reasons for a system's decision on a single specific instance. In contrast, *global explanations* operate on a larger scale, either by covering a large set of examples or by attempting to approximate the overall internal logic of the ML model (Bodria *et al*., 2023).

Among various explanation techniques, such as feature importance (Ribeiro *et al*., 2016; Lundberg and Lee, 2017) and counterfactuals (Stepka *et al*., 2024), prototypes are particularly noteworthy. Since they are actual examples of training data, prototypes are inherently easier for humans to understand compared to more abstract explanation methods (Menis Mastromichalakis *et al*., 2024). They can serve a dual role: they can function as a *local explanation* by linking a model's prediction to the most similar training instances, or as a *global explanation* by using a limited number of representative examples to approximate the model's overall decision boundaries.

Although prototypes can be applied to various data types, this paper focuses on tabular data, where examples are represented as vectors of (feature, value) pairs. However, interpreting prototypes remains challenging with many features (Menis Mastromichalakis *et al*.,

*Corresponding author

2024). For local explanations, users may struggle to determine which features are the most important for a specific prediction. For global explanations, it is desirable that the discovered prototypes be not only well-distributed across the data space but also represent diverse subsets of important feature values. This goal shares motivations with related fields, such as the mining of interpretable characteristic rules or sub-group discovery (Bach, 2025; Alkhatib *et al*., 2024).

Similar efforts have been made to improve the human interpretability of prototypes for other data modalities. For example, in image analysis, *prototypical parts networks* were introduced to identify characteristic patches rather than complete images (Chen *et al*., 2019). Similarly, for text classification, adaptations of these networks can highlight the most important phrases instead of complete text paragraphs (Pluciński *et al*., 2021). The concept has also been applied to time-series, where prototypes are defined as representative shapes of important segments (Bobek and Nalepa, 2025). However, for tabular data, the decomposition of prototypes into meaningful parts remains underexplored.

To bridge this gap, we propose a method to identify the most important features within prototypes. Our approach applies a model-agnostic explanation method to compute feature importance for the black-box model, providing a more refined perspective than existing techniques. These identified feature subsets can be used for local and global explanations, helping users better interpret the provided results.

Our method uses feature importance in two ways. First, we identify *alike parts* by highlighting the most informative overlapping features between a predicted instance and its nearest prototype. This approach directs the user's attention to a limited set of key features when interpreting a local model prediction. Second, we incorporate feature importance into the prototype selection objective function (used within the prototype generation algorithms) to promote diversity. This diversity aids in identifying meaningful alike parts. These strategies balance interpretability and diversity, enhancing both local explanations and the prototype selection process. The methods are evaluated on benchmark datasets.

This paper significantly extends our preliminary work (Karolczak and Stefanowski, 2025a). The new contributions include (1) evaluating multiple prototype generation algorithms alongside a new evaluation function, (2) exploring a larger set of feature importance algorithms and feature selection operators for the final subsets, and (3) conducting a more extensive experimental analysis, assessing the impact of the proposed method's individual components and parameters.

2. Related work

Below, we briefly review relevant work on prototype explanations and then on evaluating feature importance in black box models.

We consider a dataset $\mathcal{S}$ composed of n learning examples, formally expressed as $\mathcal{S} = (\mathbf{x}_i, y_i)_{i=1}^{n}$. In this formulation, $\mathbf{x}_i \in \mathbb{X}^d$ is a feature vector that spans d dimensions and $y_i \in \mathcal{Y}$ is its associated label. The data format assumed throughout this work is tabular, presented as feature-value pairs. Central to our approach is a trained classifier h, which works as a black-box model for prediction. This classifier h maps an input instance $\mathbf{x}_i$ to a predicted label $\hat{y}_i$, i.e., as the mapping $h(\mathbf{x}_i) \mapsto \hat{y}_i$.

A *prototype* is an instance chosen from the dataset that serves as a representative example, specifically $(\mathbf{x}_j, \hat{y}_j)$, where $\hat{y}_j$ is the class assignment yielded by the classifier h. The resulting collection of prototypes, denoted by $\mathcal{P}$, constitutes a restricted subset of the entire dataset $\mathcal{S}$. Thus, $\mathcal{P} = \{(\mathbf{x}_j, y_j)\}_{j=1}^{m}$, where the condition $m \ll n$ ($\mathcal{P} \subset \mathcal{S}$) ensures that the final number of prototypes remains significantly smaller than the initial size of the training dataset.

Various methods for creating prototype explanations have been proposed up to now. Some rely on techniques such as vector quantization and kernel functions (Schleif *et al*., 2011), sharing conceptual links with KNN-based approaches. For example, IKNN_PSLFW (Zhang *et al*., 2022) first divides the data into class-specific groups before selecting prototypes that are maximally distant from instances belonging to other classes. A critical challenge for many algorithms is their reliance on standard distance measures computed directly in the original feature space. This mandates a careful definition of similarity that can handle diverse data types (binary, numerical, categorical) and maintain robustness against scaling differences among features.

To overcome the limitations of distances in the raw feature space, recent approaches have focused on locating prototypes based on their proximity within a new representational space linked directly to the predictions of the black-box model. This shift is exemplified by the tree-space prototypes developed to explain ensemble models. They use a different view on evaluating classified instances based on their predictions; in the case of tree ensembles, the similarity of examples is determined by the degree of common decision leaves in the component trees. In this new space of that similarity, various methods of selecting representatives can be applied, drawing inspiration mainly from clustering algorithms. The initial one in this line of work, SM-A (Tan *et al*., 2020), identifies prototypes as medoids in this novel prediction space, although it requires the user to pre-specify the desired number of prototypes. A later approach, A-PETE (Karolczak and Stefanowski, 2024), subsequently extends

this process by providing a mechanism for autonomous prototype selection, using greedy medoid algorithms optimized with a specialized evaluation function (see definitions in Sections 3.2). The experiments show satisfactory agreement between the surrogate classifier using 1-NN nearest prototype similarity and the prediction of the approximated tree ensemble (Karolczak and Stefanowski, 2024).

Despite the aforementioned research on prototypes for tabular data, surprisingly few studies have addressed the optimal manner of presenting these prototypes to end-users. Although visualizations like 2D scatter plots or self-organizing maps have been proposed (Biehl *et al*., 2016), these are typically only suitable for low-dimensional data and ultimately fail to direct the user's focus toward the most critical parts of the prototype.

The concept of exposing only a subset of an instance or prototype to the user to reduce cognitive load is not novel. This practice is widely adopted in image processing, notably by ProtoPNet (Chen *et al*., 2019) and its subsequent extensions, which present specific regions or parts of images. This idea has also been adapted for tabular data, as seen in applications such as medical laboratory testing (Karolczak and Stefanowski, 2025b). However, these existing approaches typically necessitate the use of highly specific deep learning architectures. Consequently, they are not readily applicable to other powerful predictive models, such as tree ensembles, which frequently represent the state-of-the-art solution for many real-world problems.

When assessing the importance of features and their selection, it should be noted that this issue has been extensively studied in knowledge discovery in databases (KDD) and ML for decades, but mainly at the level of assessing the global importance of features in the data itself and their interdependence with the target decision (see Li *et al*., 2017; Liu and Motoda, 2000). Usually, statistical or other methods derived from information theory and wrappers around classifiers are exploited.

However, in our approach, we need specialized methods for assessing the importance of features for ML model predictions for individual instances. According to reviews (Bodria *et al*., 2023), this can be achieved by a number of proposals based on different principles. Here we direct our interest to the popular SHAP family of methods for approximate Shapley values, such as Tree SHAP (Lundberg and Lee, 2017), which generates a local explanation in the form of a vector whose length corresponds to the number of features. The values within this vector attribute an importance score to each individual feature, which helps users understand the model's behavior for a specific instance. The spectrum of feature importance techniques is broad. Some methods exploit the inherent structure of the model itself. For example, Tree Interpreter (Li *et al*., 2019) capitalizes on the architecture of tree-based models, which allows it to run rapidly. In contrast, other techniques are entirely model-agnostic, relying solely on observing the impact of perturbations to the input data on the model's output decision. These methods, exemplified by LIME and Kernel SHAP (Lundberg and Lee, 2017), are computationally expensive and therefore operate very slowly, as they do not analyze the internal workings of the predictor at all.

Although numerous methods exist to assess the influence of features on predictions made by black-box models, these techniques have been underexploited in combination with prototype-based explanations.

3. Proposed method

This section outlines the main methodological contributions of this paper. Specifically, Section 3.1 presents a method for guiding user attention in local explanations of decisions for an instance and its nearest prototype. Section 3.2 introduces a general approach to improving prototype-selection algorithms to produce prototypes that are better tailored to guide user attention.

The proposed method is formalized in a quite general way and can be used with various algorithms for constructing the prototypes. For example, in our experiments, we will consider algorithms such as A-Pete or SM-A in relation to tree ensembles.

3.1. Identifying alike parts. As noted by Menis Mastromichalakis *et al*. (2024), a prototype can often be challenging to be interpreted as a whole (with all feature value pairs), which in turn complicates explaining the prediction of a black-box model. Individual features within the prototype may contribute unequally, some playing a important role in the prediction being interpreted, while others have little influence.

We introduce a method to pinpoint the most informative features that a predicted instance shares with its prototype, highlighting the focused feature subset for the user's attention. These shared features, which we call *alike parts*, consist of feature subsets whose importance remains consistently high in both the instance and its nearest prototype.

For this aim, we propose alike part identification (see Algorithm 1), which outputs a binary mask highlighting the features the user should focus on when comparing an instance $\mathbf{x}_i$ with its nearest prototype $\mathbf{p}_j$.

Identifying alike parts starts by computing *feature importance scores* $\phi(h, \mathbf{x}_i^l)$ and $\phi(h, \mathbf{p}_j^l)$ for each feature $l \in \{1, \ldots, d\}$ using the classifier h. The method assumes that each feature is assigned exactly one score, positive or negative, without any normalization requirement. Hence, any feature importance estimation technique can be employed. For random forests, for

Table 1. Demonstration of alike parts identification for a sample instance and its nearest prototype within the Apple Quality dataset. The table illustrates the process of generating the binary feature mask $\mathbf{m}$. The first and second rows display the raw feature-importance scores (ϕ) for the instance and the prototype, respectively. The third row shows the resulting weight vector ($\mathbf{w}$), obtained using the Hadamard product as the similarity operator ($\circ$) on the normalized feature importance scores (as detailed in Section 3.1). The final row, the binary mask ($\mathbf{m}$), depicts the most relevant shared features (here, where the weight exceeds the mean threshold), denoted by the value '1'.

	Size	Weight	Sweetness	Crunchiness	Juiciness	Ripeness	Acidity
Instance	–2.77	–1.08	–1.72	1.38	0.19	3.65	0.31
Prototype	–0.97	–0.20	–3.07	0.00	-0.52	3.16	–0.52
Weights	0.18	0.02	0.27	0.00	0.00	0.51	0.00
Mask	1	0	1	0	0	1	0

Algorithm 1. Alike parts identification.

Input: classifier h, instance to be explained $\mathbf{x}_i$, and its nearest prototype $\mathbf{p}_j$
$\mathbf{f}_x, \mathbf{f}_p \leftarrow \text{get_fi}(h, \mathbf{x}_i), \text{get_fi}(h, \mathbf{p}_j)$
if normalize_similarity **then** normalize($\mathbf{f}_x, \mathbf{f}_p$)
if ignore_direction **then** $\mathbf{f}_x, \mathbf{f}_p \leftarrow |\mathbf{f}_x|, |\mathbf{f}_p|$
$\mathbf{w} \leftarrow \mathbf{f}_x \circ \mathbf{f}_p$
mask $\leftarrow$ weights_to_mask($\mathbf{w}$)
Output: mask

example, one may use Tree SHAP (Lundberg and Lee, 2017) or Tree Interpreter (Li *et al.*, 2019). Note that this is the most computationally expensive step of the proposed method. Therefore, while any local feature importance estimation technique can be utilized, prioritizing one with high computational efficiency is advised.

Optionally, the user can ignore the direction of feature importance scores ('ignore_direction') by taking absolute values or normalize the scores ('normalize_similarity'). The ignoring direction is defined as the simple absolute value of the raw score:

$$\hat{\phi}(h, \mathbf{x}_i^l) = |\phi(h, \mathbf{x}_i^l))| \,. \tag{1}$$

Optional scores normalization is defined as follows:

$$\hat{\phi}(h, \mathbf{x}_i^l) = \frac{(\phi(h, \mathbf{x}_i^l))^2}{\sum_{k=1}^{d} (\phi(h, \mathbf{x}_i^k))^2} \,. \tag{2}$$

The alignment of feature importance between the instance and its prototype is expressed as the similarity operator ($\circ$) applied to their respective scores:

$$w^l = \hat{\phi}(h, \mathbf{x}_i^l) \circ \hat{\phi}(h, \mathbf{p}_j^l) \,. \tag{3}$$

In this work, we consider several options for the $\circ$ similarity operator. These operators are applied element-wise to produce the weight vector $\mathbf{w}$ (where $w^l = \hat{\phi}(h, \mathbf{x}_i^l) \circ \hat{\phi}(h, \mathbf{p}_j^l)$):

- Hadamard (element-wise): $w^l = \hat{\phi}(h, \mathbf{x}_i^l) \cdot \hat{\phi}(h, \mathbf{p}_j^l)$,
- element-wise l_1: $w^l = 1 - |\hat{\phi}(h, \mathbf{x}_i^l) - \hat{\phi}(h, \mathbf{p}_j^l)|$,
- element-wise l_2: $w^l = 1 - (\hat{\phi}(h, \mathbf{x}_i^l) - \hat{\phi}(h, \mathbf{p}_j^l))^2$.

We select these measures because they are widely used in loosely related works (in particular, by Chen *et al.* (2019)).

The computed weights are critical, as they quantify each feature's importance in driving the model's prediction for both the nearest prototype and the classified instance. Based on the values of these weights, a subset of features is selected by constructing a binary feature mask $\mathbf{m} \in \{0, 1\}^d$. This selection process can be carried out in various ways. Below we propose the following strategies based on different principles:

1. *Mean thresholding.* The simplest approach retains any feature l whose corresponding weight w^l exceeds the average weight across all d features:

$$m^l = \mathbb{1}\left(w^l > \frac{1}{d}\sum_{k=1}^{d} w^k\right) \,. \tag{4}$$

2. *Adaptive top-k selection.* To achieve explanations with controlled sparsity, we explore the selection of the k features that possess the highest weights. We specifically test two data-driven scales for k: selecting the top $k = \lceil\sqrt{d}\rceil$ features and selecting the top $k = \lceil\log(d)\rceil$ ones (they are inspired by Breiman's research on feature selection in Random Forest (Breiman, 2001)).

In addition, the kneed algorithm (Satopaa *et al.*, 2011) was also evaluated to identify the optimal feature-set size. However, it was dismissed because its hyperparameter sensitivity resulted in unstable outcomes, making it unreliable.

Table 1 serves to illustrate the initial selection procedure using the mean-based threshold.

3.2. Augmented objective for prototype selection. The algorithms established for prototype selection, including those presented by Tan *et al.* (2020) or Karolczak and Stefanowski (2024), frame the problem of

determining representative data points as a variation of the problem of k-medoids. This typically relies on a greedy approximation approach. The conventional k-medoids objective is to minimize a distance function d between each training example $\mathbf{x}_i$ and its nearest prototype $\mathbf{p}_j$. This objective is standardly expressed as

$$f(\mathcal{P}) = \sum_{i=1}^{|\mathcal{S}|} \min_{\mathbf{p}_j \in \mathcal{P}} d\left(\mathbf{x}_i, \mathbf{p}_j\right), \tag{5}$$

where $|\mathcal{S}|$ denotes the cardinality of the training set. The specific distance function d utilized depends on the underlying architecture of the model. For example, in neural networks, this may involve a dot product of trainable embeddings (Li *et al.*, 2018) or, for tree ensembles, a specialized tree distance metric (Tan *et al.*, 2020; Karolczak and Stefanowski, 2024).

To diversify the influence of features in different prototypes, we propose extending the objective function by incorporating a term based on the feature importance score, fi. This component is derived from the similarity operator ($\circ$) defined in Section 3.1 and applied to the pre-processed feature importance scores ($\hat{\phi}$).

The feature importance score component, fi, is defined as the sum of the resulting weights w^l (from Eqn. (3), where $w^l = \hat{\phi}(h, \mathbf{x}_i^l) \circ \hat{\phi}(h, \mathbf{p}_j^l)$) for the l-th feature of instance $\mathbf{x}_i$ and its closest prototype $\mathbf{p}_j$:

$$fi(\mathbf{x}_i, \mathbf{p}_j) = \sum_{l=1}^{d} w^l = \sum_{l=1}^{d} \left(\hat{\phi}(h, \mathbf{x}_i^l) \circ \hat{\phi}(h, \mathbf{p}_j^l)\right). \tag{6}$$

The specific functional form of fi thus depends on the chosen configuration for the $\hat{\phi}$ scores (i.e., whether the direction is ignored or normalization is applied) and the selected similarity operator $\circ$ (i.e., the Hadamard product, element-wise $1 - l_1$, or element-wise $1 - l_2$).

The fi scores across all $\mathbf{x}_i$ can be pre-calculated and stored efficiently before the optimization procedure begins. The reformulated objective function integrates the classical distance minimization with an adjustable term, weighted by the parameter β, which accounts for the defined feature importance score. The first term ensures that the dataset $\mathcal{S}$ is compactly covered by the prototypes, as each instance is assigned to nearest representative. The second β-weighted term promotes a wider representation of feature importance among the selected prototypes. The new optimization objective is formally presented as

$$f(\mathcal{P}) = \sum_{i=1}^{|\mathcal{S}|} \min_{\mathbf{p}_j \in \mathcal{P}} \left(d\left(\mathbf{x}_i, \mathbf{p}_j\right) + \beta \cdot fi\left(\mathbf{x}_i, \mathbf{p}_j\right)\right). \tag{7}$$

This modification facilitates a more sophisticated global prototype selection process, where the parameter β acts as a trade-off mechanism between the metric distance and feature importance alignment. This updated formulation ultimately leads to specilized prototype selection for identifying alike parts.

A key advantage of the proposed method is its inherent robustness when handling missing values. This resilience relies on the native capabilities of the underlying components. Since we employ prototype selection algorithms (Tan *et al.*, 2020; Karolczak and Stefanowski, 2024) using Random Forest (RF) (Breiman, 2001), along with feature estimation methods such as Tree SHAP and Tree Interpreter, all of which natively support missing values, the entire method does not require additional preprocessing steps for missing data.

The augmented prototype selection scales quadratically, $\mathcal{O}(n^2)$, with the number of training instances n, driven by the pairwise distance and feature importance computations. Conversely, the online inference phase is highly efficient; it scales linearly, $\mathcal{O}(m)$, with the number of selected prototypes m. Because $m \ll n$, the resulting surrogate model maintains strictly bounded, compact memory usage.

4. Experiments

The experiments aim to (1) demonstrate the usefulness of the proposed method in terms of identifying important subsets of features for selected datasets and (2) study the impact of its components and hyperparameters.

This part first details the setup (Section 4.1), followed by a case study on alike parts (Section 4.2), and concludes with an analysis of the surrogate model's performance (Section 4.3) as well as the number of activating features (Section 4.5).

4.1. Experimental setup. The proposed framework was implemented in Python using standard machine learning libraries, including scikit-learn[1] for model training, SHAP[2] and Tree Interpreter[3] for feature importance estimation, and NumPy[4] for custom functions and vectorized operations. The compact, modular codebase and reproducible experiment scripts are publicly available on GitHub[5]. Empirically, for the evaluated datasets, the prototype selection pipeline executed in the order of seconds on a personal computer (Apple M2, 16 GB RAM). The resulting 1-NN surrogate models maintain minimal memory usage and provide highly efficient inference, demonstrating execution times comparable to the standard baselines.

[1] Available online at `https://scikit-learn.org`.

[2] Available online at `https://shap.readthedocs.io`.

[3] Available online at `https://github.com/andosa/treeinterpreter`.

[4] Available online at `https://numpy.org`.

[5] Available online at `https://github.com/jkarolczak/alike-parts`.

4.1.1. Algorithms. As detailed in Section 3.2, our augmented optimization method is broadly applicable to various algorithms. For this study, we specifically implemented this modification in three different prototype selection algorithms that rely on optimizing a specialized tree distance metric (based on the degree of achieving the same decision tree leaves) to explain the Random Forest ensemble (Breiman, 2001), a type of multiple classifier system known to perform well across many applications (Woźniak *et al*., 2014). These three algorithms were chosen because they represent diverse strategies for prototype selection within the tree distance space and could be easily augmented with the proposed extension of the objective function:

- *G-KM* selects an equal, predetermined number of prototypes within each class (greedy k-medoid approximation computed within classes);
- *SM-A* (Tan *et al*., 2020) selects the prototype that yields the largest marginal improvement in the objective function across all classes;
- *A-PETE* (Karolczak and Stefanowski, 2024) automates prototype selection based on relative improvement thresholds (for details, see the work of Karolczak and Stefanowski (2024)).

4.1.2. Datasets. We utilize six benchmark datasets for evaluation, varying in dimensionality from seven to 31 features. These datasets are chosen to represent two distinct feature importance profiles, enabling us to test the robustness of our selection methods under diverse feature sparsity conditions. Four datasets possess a subset of globally important features: Australia Rain, Breast Cancer, Diabetes, and Passenger Satisfaction. The remaining two, Apple Quality and Wine Quality, exhibit a high importance spread across most or all features.[6]

4.1.3. Metric. To evaluate the quality of the selected set of prototypes $\mathcal{P}$, we measure the fidelity of a surrogate model built from these prototypes. This approach is consistent with prior work (Tan *et al*., 2020; Karolczak and Stefanowski, 2024).

Following the evaluation procedure by Tan *et al*. (2020), our *surrogate model* is a 1-nearest neighbor (1-NN) classifier that uses the selected set of prototypes $\mathcal{P}$ as its training data. The 1-NN classifier assigns to a test instance $\mathbf{x}_{test}$ the label of its nearest prototype $\mathbf{p}_j \in \mathcal{P}$, where the nearest is determined by the same specialized tree distance metric d used in the selection process.

Fidelity is then defined as the accuracy of this 1-NN surrogate model in matching the predictions of the original black-box model on a hold-out test set $\mathcal{S}_{test}$. Fidelity measures how well the small set of prototypes can represent the underlying black-box model's decision boundaries for classifying new, unseen data.

4.2. Case study: Analyzing alike parts. The mechanism for identifying alike parts is demonstrated as an instructional example in Table 1, which details the computation of weights from the feature importance vectors of an instance and its prototype.

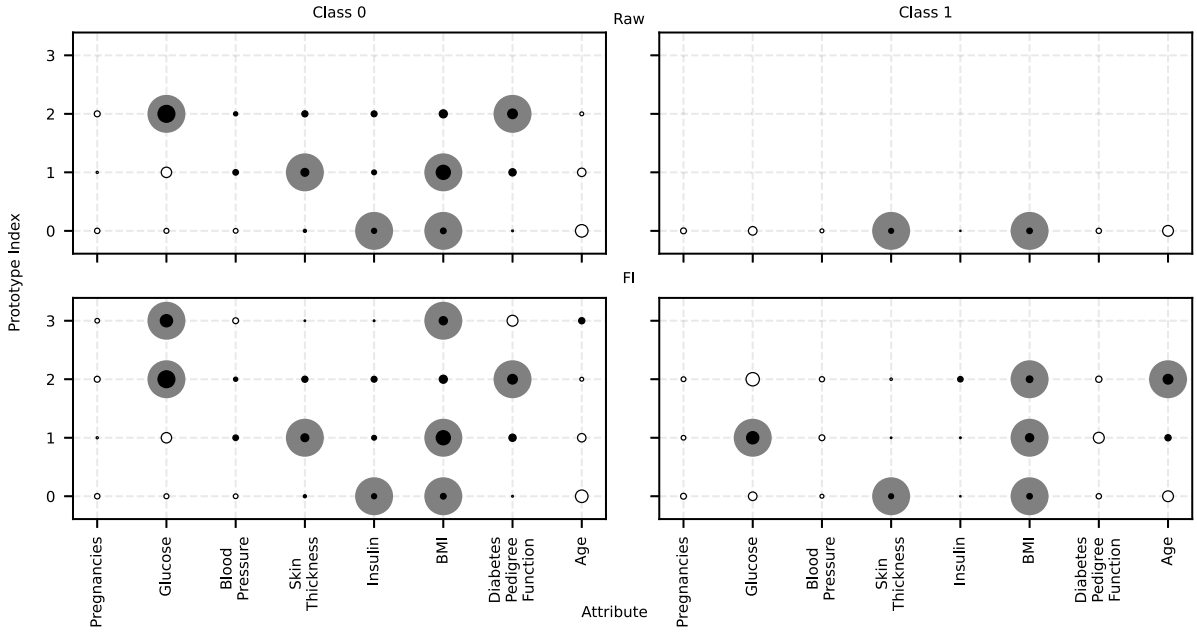
Furthermore, Table 2 presents a comparison of the selected alike parts for two instances and their nearest prototype, contrasting the original (raw) and the feature importance (FI)-informed versions of the A-PETE algorithm on the Diabetes dataset. Integrating the feature importance component into A-PETE's optimization resulted in a noticeably different set of prototype selections than in the raw algorithm when operating with a black-box RF (Breiman, 2001).

The prototype chosen by the raw A-PETE only identifies *Glucose* as a feature important to both the instance and the prototype. In contrast, the FI-informed algorithm additionally highlights *Diabetes Pedigree Function* and *Age*. This outcome is consistent with the established medical literature on diabetes risk factors (Kautzky-Willer *et al*., 2016). This result illustrates our method's capacity to facilitate the discovery of more relevant and meaningful relationships between target instances and their representative prototypes.

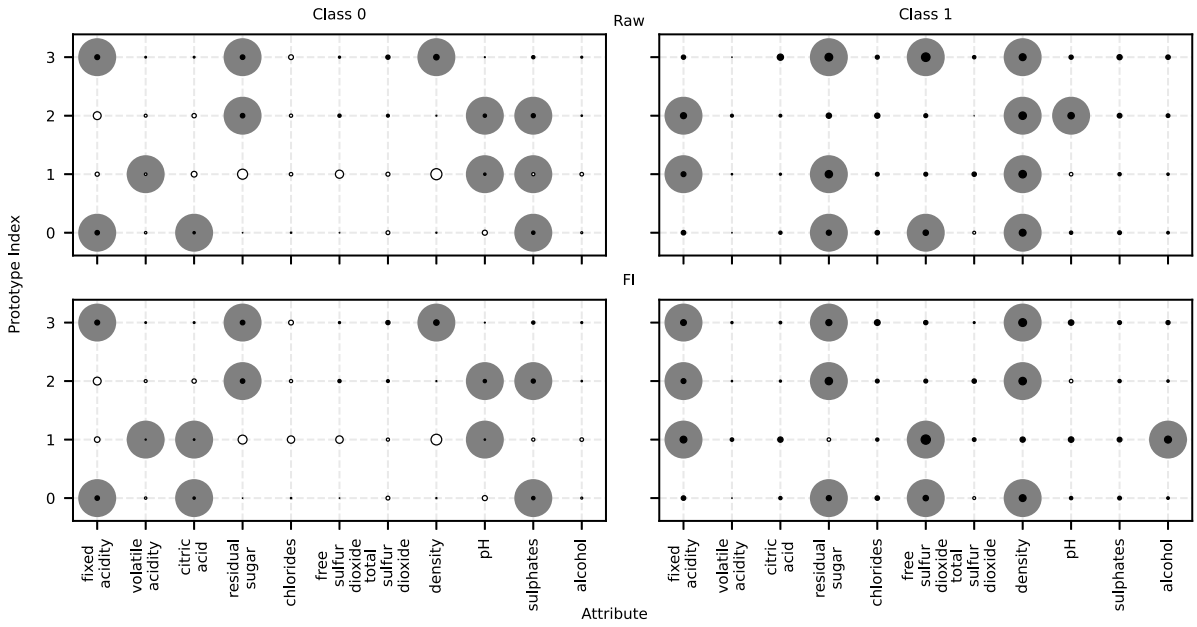
Beyond this single-instance analysis, a visual comparison of the globally generated sets of prototypes and their selected important features for the Diabetes and Wine Quality datasets is presented in Fig. 1. This figure directly contrasts the prototypes produced by the original (raw) algorithms with those generated using the feature importance-informed approach. The visualization confirms that the FI-informed algorithm generates prototypes with greater diversification, effectively highlighting feature subsets that vary between the selected representatives.

For example, Fig. 1(a) illustrates that the *Age* feature was selected as an important feature in at least one prototype only when FI was incorporated into the objective function. Figure 1(b) depicts a comparable scenario for the Wine Quality dataset, where *alcohol* is selected as important. This identical phenomenon was also observed for the Australia Rain and Breast Cancer datasets, where certain features were deemed

[6]All datasets are sourced from `https://www.kaggle.com/datasets/`, while their respective online paths are `https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package`, `https://www.kaggle.com/datasets/rahmasleam/breast-cancer`, `https://www.kaggle.com/datasets/mathchi/diabetes-data-set`, `https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction`, `https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality`, and `https://www.kaggle.com/datasets/taweilo/wine-quality-dataset-balanced-classification`.



(a) prototypes and features identified as important using the A-PETE algorithm and Tree Interpreter on the Diabetes dataset; the binary feature mask $\mathbf{m}$ was generated from the weight vector $\mathbf{w}$ by selecting $k = \lceil \log d \rceil$ features.



(b) prototypes and features identified as important using the G-KM algorithm and Tree SHAP on the Wine Quality dataset; the binary feature mask $\mathbf{m}$ was generated from the weight vector $\mathbf{w}$ by selecting $k = \lceil \sqrt{d} \rceil$ features.

Fig. 1. Comparison of prototypes (x-axis: prototype index) and important features (y-axis: feature index). The top row displays prototypes generated using the original raw algorithm, while the bottom row incorporates the augmented target function with feature importance (FI). The size of the inner circle represents the magnitude of the feature importance, and a gray highlight denotes features identified as important for a given prototype. The parameters employed in this analysis were selected to maximize fidelity, consistent with the results presented in Table 3.

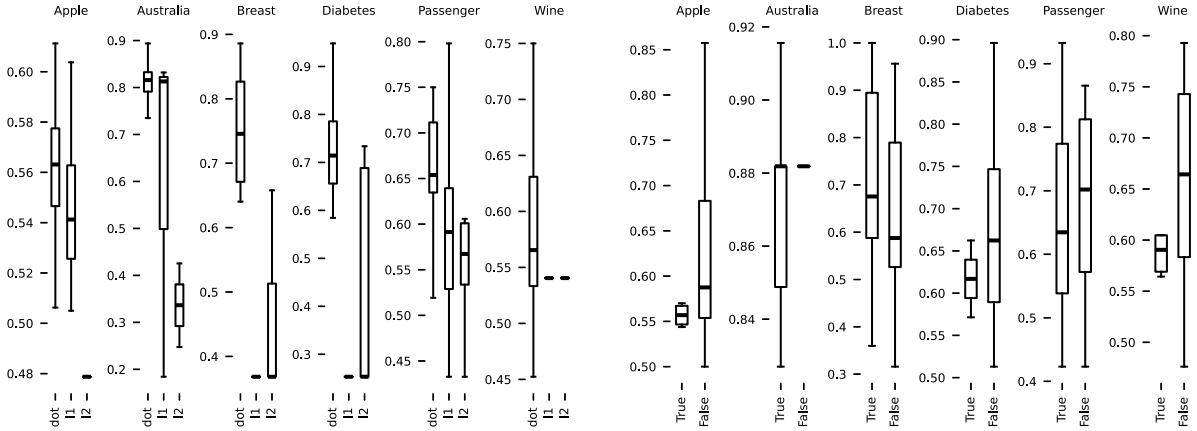


Fig. 2. Fidelity comparison of prototypes found with augmented objective across various configurations. The boxplots illustrate the distribution of fidelity across datasets; subfigure (a) compares the impact of similarity metrics for SM-A and Tree SHAP; subfigure (b) compares the effect of ignoring the direction of feature importance (applying absolute value) for APete and Tree Interpreter.

Table 2. Example of alike parts identified for an instance and its nearest prototype using the A-PETE algorithm (Karolczak and Stefanowski, 2024). The comparison illustrates feature selection results based on two distinct prototype optimization definitions: the original (raw) method and the feature importance informed approach utilizing Tree SHAP. Features identified as alike between the explained instance and the prototype are distinguished by formatting: the FI-informed alike parts are displayed in bold, and alike parts selected in the prototypes from original (raw) problem formulation are italicised.

Class	Type	Pregnancies	Glucose	BloodP.	SkinT.	Insulin	BMI	PedigreeF.	Age
0	Instance	6	***102***	82	0	0	30.8	**0.18**	**36**
	Prototype (FI)	7	**125**	86	0	0	37.6	**0.30**	**51**
	Prototype (Raw)	7	*62*	78	0	0	32.6	0.39	41
1	Instance	***8***	***100***	74	40	215	39.4	**0.66**	43
	Prototype (FI)	**9**	**152**	78	34	171	34.2	**0.89**	33
	Prototype (Raw)	*9*	*171*	110	24	240	45.4	0.72	54

important exclusively by the FI-informed version of the algorithm. Furthermore, augmenting the G-KM algorithm with FI—an algorithm designed to find a predefined number of prototypes—results in a more diversified set of prototypes, which is particularly observable in the prototypes selected for class 1 in Fig. 1(b).

4.3. Surrogate model performance. The predictive performance of the surrogate 1-NN model, which is constructed from the generated prototypes, is quantified using fidelity. This evaluation is consistent with the established methodology found in related literature (Tan *et al.*, 2020; Karolczak and Stefanowski, 2024). This surrogate model classifies instances from a hold-out test set by employing a 1-nearest neighbor (1-NN) search within the subset of selected prototypes. Our primary focus here is to quantify the impact of our modified prototype selection approach, which integrates feature importance, on the resulting model's accuracy. All hyperparameter settings were determined through extensive hyperparameter tuning using Bayesian optimization implemented via Optuna[7] and Tree-Structured Parzen Estimator (Watanabe, 2025).

The maximal fidelity results for both the original (raw) and the FI-augmented prototype selection algorithms are summarized in Table 3. As baselines, we evaluate three interpretable proxy explainers: Naive Bayes, Logistic Regression, and Decision Tree (capped at depth 15 to balance capacity and interpretability). These are trained on the black-box Random Forest predictions to approximate its decision boundaries, using mean imputation for missing data.

It can be observed that integrating feature importance into the objective function yields highly competitive fidelity. A Wilcoxon signed-rank test comparing the original prototype selection against the proposed feature-informed objective indicates a positive trend toward improved fidelity ($p \approx 0.068$). Furthermore, evaluating the proposed prototype-based 1-NN explainers

[7] Available online at `https://optuna.org`.

Table 3. Comparison of fidelity of prototypes in 1-NN search selected using three different configurations for the feature importance (FI) term in the augmented objective function. Interpretable models (NB: Naive Bayes, LR: Logistic Regression, DT: Decision Tree) trained on black-box predictions are included as baselines.

Dataset	Baseline fidelity			Prototype fidelity			
	NB	LR	DT	Method	Tree SHAP	Tree Interpreter	Raw
Apple Quality	0.811	0.809	0.833	A-Pete	0.864	0.858	0.787
				G-KM	0.949	0.950	0.932
				SM-A	0.611	0.846	0.787
Australia Rain	0.859	0.932	0.835	A-Pete	0.892	0.916	0.930
				G-KM	0.935	0.963	0.956
				SM-A	0.887	0.899	0.893
Breast Cancer	0.947	0.939	0.921	A-Pete	0.965	1.000	1.000
				G-KM	1.000	1.000	1.000
				SM-A	0.851	0.956	1.000
Diabetes	0.890	0.890	0.818	A-Pete	0.864	0.896	0.903
				G-KM	0.955	0.916	0.909
				SM-A	0.948	0.831	0.883
Passenger Satisfaction	0.913	0.894	0.885	A-Pete	0.962	0.933	0.962
				G-KM	0.952	0.952	0.952
				SM-A	0.856	0.856	0.942
Wine Quality	0.705	0.655	0.705	A-Pete	0.786	0.769	0.760
				G-KM	0.867	0.829	0.798
				SM-A	0.786	0.724	0.769

against the standard predictive baselines demonstrates a significant improvement in fidelity ($p < 0.05$).

4.4. Sensitivity analysis. Figure 2 examines the influence of two key hyperparameters on the fidelity of the resulting model. Figure 2(a) shows that, for Tree SHAP and SM-A, the Hadamard product as a similarity metric works best across all datasets. In general, a specific hyperparameter value can show a global preference by consistently yielding higher fidelity across all evaluated datasets. However, as observed in Fig. 2(b) for the 'ignore_direction' hyperparameter, while there is occasionally some trend, tailored tuning remains necessary, as there are usually datasets that do not match the global trend. The complete set of figures for all hyperparameters is available in the electronic appendix[8].

4.5. Activating features. The proposed approach was validated by analyzing feature activation frequencies on the test sets. Figure 3 shows that these frequencies differ notably between the original (raw) and FI-informed strategies. For the Apple Quality dataset (A-PETE with Tree SHAP), activation frequencies vary less when $\beta = 2.5$ compared to $\beta = 0.5$. This reduced variance demonstrates that the FI-informed objective successfully promotes feature diversity. Explicitly penalizing repeated feature usage forces the selection of a more diverse

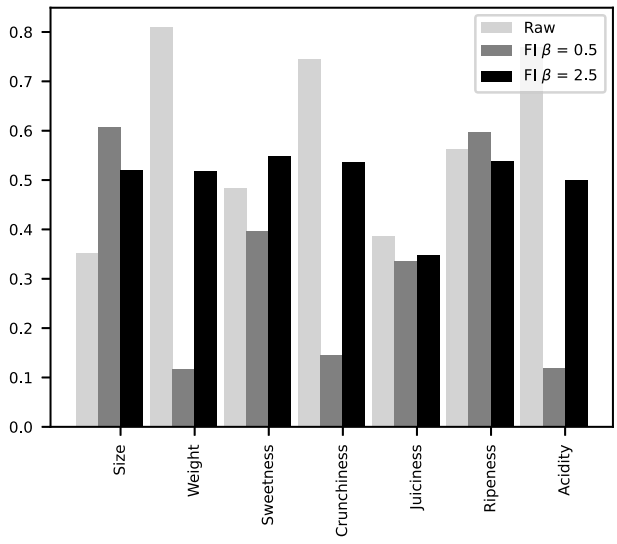


Fig. 3. Comparison of the frequency of feature highlighting between the original (raw) and FI-informed strategies (with two beta values) for the Apple Quality dataset. The prototypes were found using the A-PETE algorithm and Tree SHAP, with the mean threshold as the masking strategy.

[8]Available online at `https://jkarolczak.github.io/appendix-alike-parts`.

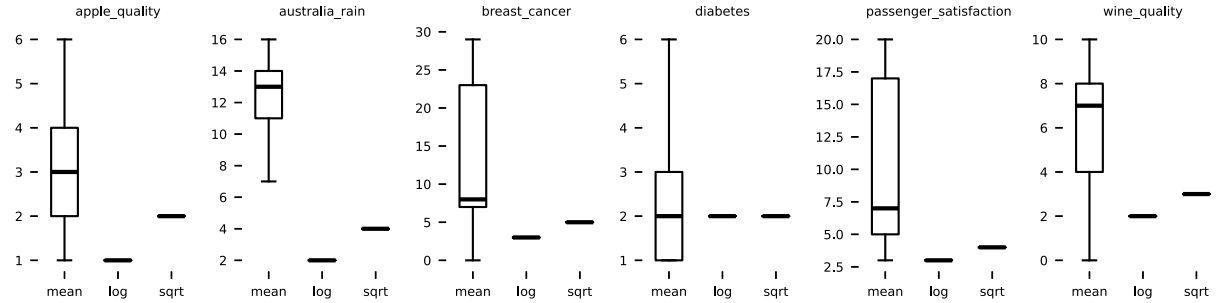


Fig. 4. Statistics for the number of features identified as alike parts across all datasets. The results were compiled from all algorithm and masking combinations, using the specific hyperparameters that maximized test set fidelity (as reported in Table 3).

set of prototypes, ensuring broader coverage of feature influence.

Figure 4 provides a quantitative comparison of the number of features included in the identified alike parts across various experimental configurations, including different datasets, prototype selection algorithms, feature importance estimators, and feature masking strategies.

A clear trend emerges regarding the length of the resulting feature subset: the mean thresholding strategy typically generates the longest alike parts. This is followed sequentially by the two top-k approaches: $k = \lceil\sqrt{d}\rceil$ and finally the most restrictive $k = \lceil\log d\rceil$.

Crucially, the mean thresholding method offers the greatest inherent flexibility because its threshold is data-dependent, allowing it to dynamically adapt the resulting length of the alike part to the specific characteristics of each instance-prototype pair.

5. Discussion

This paper addressed a key limitation of prototype-based explanations: their failure to highlight which specific features drive the similarity of a prototype for a given instance. We introduced the concept of alike parts by integrating feature importance directly into the prototype selection objective. This dual approach enhances local explanations by focusing user attention while simultaneously improving the global set of prototypes by promoting feature diversity.

A primary finding is that augmenting the prototype selection objective with a feature importance term, a move intended to boost feature diversity, does not necessarily degrade model fidelity. As shown in Section 4.3, this augmentation frequently preserved or even increased the fidelity of the 1-NN surrogate model. This suggests that seeking a diverse set of feature attributions can be complementary to the distance-based coverage requirement. This global optimization for feature diversity also yielded qualitatively superior local explanations. For example, in the Diabetes case study (Section 4.2), the feature-informed method successfully identified prototypes that share clinically relevant factors such as *Age* and *Pedigree Function*, connections that the baseline distance-only method missed.

A key aspect of our framework is the adaptability of the augmented objective function (Eqn. (7)). The hyperparameter β (analyzed in Section 4.5) provides explicit control over the trade-off between traditional distance-based coverage (d) and feature importance alignment (fi). Although our experiments utilized a tree-specific metric for Random Forest, the framework's modular, model-agnostic design allows easy substitution of these components. For instance, it can be applied to neural networks by defining d as the l_2 distance within the penultimate layer's embedding space and deriving fi using architecture-specific estimators like Deep SHAP or Integrated Gradients. Furthermore, the proposed diversity measure readily adapts to conceptually different paradigms. In clustering algorithms, it can be integrated directly into the distance metric. For instance selection, retention criteria can be modified to preserve distinct feature-level explanations. In gradient-based learning, it can serve as a regularization term in the loss function. This design allows us to extend our framework to various architectures simply by substituting the distance function d and the importance estimator fi.

Rather than modifying data, alike parts guide user attention toward shared rationale, while unshared features indicate where local explanations diverge. Eliminating unselected features alters the prototype's traversal through Random Forest; missing values can redirect it into different leaves, potentially changing its class assignment. Retaining the full feature set ensures the explanation remains consistent with the model's internal logic.

Crucial future work is to empirically validate our central hypothesis, i.e., that alike parts improve user comprehension. This claim, currently supported by proxy metrics, must be tested directly through formal human-computer interaction studies using user-centric tasks. A significant technical extension would be to adapt the framework to other black-box models, such as deep neural networks, by substituting tree-specific components

with counterparts such as latent space distances and gradient-based attribution methods for fi.

Acknowledgment

The research by Jerzy Stefanowski was funded by the National Science Centre in Poland through the OPUS grant no. 2023/51/B/ST6/00545. The research by Jacek Karolczak was funded by the Poznan University of Technology SBAD grant no. 0311/SBAD/0770.

References

Alkhatib, A., Boström, H. and Vazirgiannis, M. (2024). Explaining predictions by characteristic rules, *in* M.R. Amini *et al.* (Eds), *Machine Learning and Knowledge Discoovery in Databases*, Springer, Berlin/Heidelberg, pp. 389–403, DOI: 10.1007/978-3-031-26387-3_24.

Bach, J. (2025). Subgroup discovery with small and alternative feature sets, *ACM on Management of Data* **3**(3): 1–27.

Biehl, M., Hammer, B. and Villmann, T. (2016). Prototype-based models in machine learning, *WIREs Cognitive Science* **7**(2): 92–111.

Bobek, S. and Nalepa, G.J. (2025). TSPROTO: Fusing deep feature extraction with interpretable glass-box surrogate model for explainable time-series classification, *Information Fusion* **124**: 103357.

Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D. and Rinzivillo, S. (2023). Benchmarking and survey of explanation methods for black box models, *Data Mining and Knowledge Discovery* **37**: 1719–1778.

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

Chamola, V., Hassija, V., Sulthana, A.R., Ghosh, D., Dhingra, D. and Sikdar, B. (2023). A review of trustworthy and explainable artificial intelligence (XAI), *IEEE Access* **11**: 78994–79015.

Chen, C., Li, O., Tao, C., Barnett, A.J., Su, J. and Rudin, C. (2019). This looks like that: Deep learning for interpretable image recognition, *Conference on Neural Information Processing Systems, Vancouver, Canada*.

Karolczak, J. and Stefanowski, J. (2024). A-PETE: Adaptive prototype explanations of tree ensembles, *Progress in Polish Artificial Intelligence Research*, Warsaw University of Technology Press, Warsaw, Poland, pp. 2–8.

Karolczak, J. and Stefanowski, J. (2025a). This part looks alike this: Identifying important parts of explained instances and prototypes, *xAI 2025: Late-breaking Work, Demos and Doctoral Consortium at xAI, Istanbul, Turkey*, pp. 33–40.

Karolczak, J. and Stefanowski, J. (2025b). An interpretable prototype parts-based neural network for medical tabular data, *Proceedings of the 2nd Workshop on Second Workshop on Explainable Artificial Intelligence for the Medical Domain (EXPLIMED 2025) co-located with 28th European Conference on Artificial Intelligence (ECAI 2025), Bologna, Italy* .

Kaur, D., Uslu, S., Rittichier, K.J. and Durresi, A. (2022). Trustworthy artificial intelligence: A review, *ACM Computing Surveys* **55**(39): 1-38.

Kautzky-Willer, A., Harreiter, J. and Pacini, G. (2016). Sex and gender differences in risk, pathophysiology and complications of type 2 diabetes mellitus, *Endocrine Reviews* **37**(3): 278–316.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J. and Liu, H. (2017). Feature selection: A data perspective, *ACM Computing Surveys* **50**(6): 1–45.

Li, O., Liu, H., Chen, C. and Rudin, C. (2018). Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions, *32nd AAAI Conference on Artificial Intelligence, New Orleans, USA*, pp. 3530–3537.

Li, X., Wang, Y., Basu, S., Kumbier, K. and Yu, B. (2019). A debiased MDI feature importance measure for random forests, *in* H. Wallach *et al.* (Eds), *Advances in Neural Information Processing Systems*, Curran Associates, Red Hook, Vol. 32, pp. 8049–8059.

Liu, H. and Motoda, H. (2000). *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic, Boston.

Lundberg, S.M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions, *Conference on Neural Information Processing Systems, Long Beach, USA*, pp. 4768–4777.

Menis Mastromichalakis, O., Filandrianos, G., Liartis, J., Dervakos, E. and Stamou, G. (2024). Semantic prototypes: Enhancing transparency without black boxes, *CM Conference on Information and Knowledge Management, Boise, USA*, pp. 1680–1688.

Pluciński, K., Lango, M. and Stefanowski, J. (2021). Prototypical convolutional neural network for a phrase-based explanation of sentiment classification, *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Bilbao, Spain*, pp. 457–472.

Ribeiro, M.T., Singh, S. and Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier, *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, USA*, pp. 1135–1144.

Satopaa, V., Albrecht, J., Irwin, D. and Raghavan, B. (2011). Finding a "kneedle" in a haystack: Detecting knee points in system behavior, *31st International Conference on Distributed Computing Systems Workshops, Minneapolis, USA*, pp. 166-171.

Schleif, F.-M., Villmann, T., Hammer, B. and Schneider, P. (2011). Efficient kernelized prototype based classification, *International Journal of Neural Systems* **21**(06): 443–457.

Stepka, I., Lango, M. and Stefanowski, J. (2024). A multi-criteria approach for selecting an explanation from the set of counterfactuals produced by an ensemble of explainers, *International Journal of Applied Mathematics and Computer Science* **34**(1): 119–133, DOI: 10.61822/amcs-2024-0009.

Tan, S., Soloviev, M., Hooker, G. and Wells, M.T. (2020). Tree space prototypes: Another look at making tree ensembles interpretable, *ACM-IMS Foundations of Data Science Conference, Seattle, USA*, p. 23–34.

Watanabe, S. (2025). Tree-structured Parzen estimator: Understanding its algorithm components and their roles for better empirical performance, Available online at `https://arxiv.org/abs/2304.11127`.

Woźniak, M., Graña, M. and Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems, *Information Fusion* **16**: 3–17.

Zhang, X., Xiao, H., Gao, R., Zhang, H. and Wang, Y. (2022). K-nearest neighbors rule combining prototype selection and local feature weighting for classification, *Knowledge-Based Systems* **243**: 108451.



Jacek Karolczak is a PhD student in the Institute of Computing Science, Poznan University of Technology, from which he holds his MSc and BSc degrees in artificial intelligence. His research interests focus on interpretable machine learning, including both ante-hoc and post-hoc approaches, with a particular emphasis on prototype-based explanations and machine learning for data streams with concept drift.



Jerzy Stefanowski is a full professor in the Institute of Computing Science of the Poznan University of Technology and a member of the Polish Academy of Sciences. His research interests include machine learning, data mining and intelligent decision support, in particular rule induction, ensembles, class imbalance, concept drift, classification of data streams and explainable AI.

Received: 14 November 2025
Revised: 20 March 2026
Accepted: 2 June 2026