Z-SCORE RULE INDUCTION: AN INTERPRETABLE LIGHTWEIGHT CLASSIFIER

Anonymous authors
Paper under double-blind review

ABSTRACT

Extracting interpretable rules to retrieve rare target instances from large unlabeled populations is a critical challenge when confirmed positive cases are extremely scarce. While decision trees offer interpretability, they overfit to the small set of positive examples and fail to generalize to unseen positive instances. To address this, we propose Z-Score Rule Induction (ZSRI), a lightweight algorithm specifically designed to induce simple, interpretable rules from tiny positive seeds in a single pass. ZSRI ranks features by their positive-class deviations using univariate z-scores and induces decision boundaries via extremum- or quantile-based strategies to encompass target positive instances. Experimental evaluations on Wisconsin Breast Cancer, Pima Indians Diabetes, and synthetic benchmarks demonstrate that ZSRI achieves competitive predictive performance while executing up to $28\times$ faster than decision trees. Furthermore, on the Wisconsin Breast Cancer dataset, ZSRI trained with as few as 10 positive seed instances recovers up to 83.9% of unseen positive cases while maintaining a high precision of 96.7%.

1 INTRODUCTION

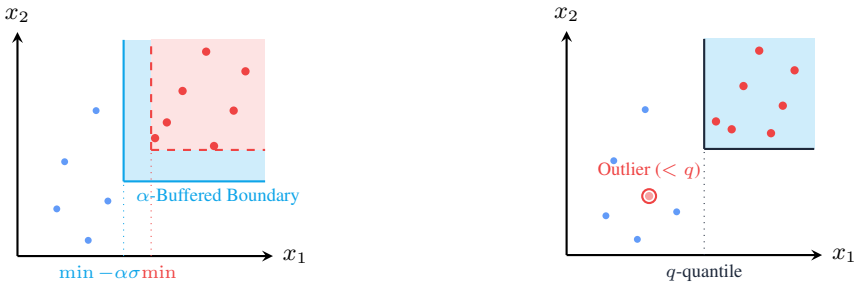
In many domains, such as clinical diagnostics, having interpretable predictions is often a requirement. Black-box models such as deep neural networks do not provide any insight into their decision making process. Furthermore, confirmed positive cases (e.g., rare diseases or biomarker signatures) are scarce relative to the normal population in these domains. Here, a primary goal is *rule discovery*: learning a simple, human-interpretable logical rule that tightly bounds target instances and retrieves unseen positive samples from the population without extracting too many false positives.

While traditional decision trees and rule-based classifiers (e.g., RIPPER) offer transparency, their recursive greedy partitioning based on impurity splits easily overfits or collapses under extreme class imbalance. Furthermore, finding such rules is generally NP-hard, making combinatorial search computationally intensive.

To address these limitations, we propose **Z-Score Rule Induction (ZSRI)**, a lightweight, non-recursive algorithm designed to encompass positive instances in a single pass. Instead of recursive partition search, ZSRI makes use of z-scores to identify important features and then derives threshold rules based on the distribution of the positive class samples. To find the threshold, two strategies are presented: an extremum-based strategy, optimized for recall, and a quantile-based strategy, optimized for precision. Geometrically, ZSRI constructs an axis-aligned bounding orthant in the feature space that encompasses positive instances, as illustrated in Figure 1. We show that ZSRI functions not only as a rapid classifier, but also as an effective tool for few-shot positive rule discovery: given as few as $s \in \{3, 5, 10\}$ positive seed examples, ZSRI induces robust logical rules that successfully retrieve remaining positive instances in unseen data while maintaining high specificity.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 formally defines the problem formulation. Section 4 details the ZSRI algorithm, its threshold strategies, and computational complexity. Section 5 presents comprehensive empirical evaluations, including scalability benchmarks and few-shot positive retrieval experiments, and Section 6 concludes the paper.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Extremum-Based Thresholding (α-Buffer) (b) Quantile-Based Thresholding (q-Quantile)

Figure 1: 2D feature space with red points representing the positive class and blue points the negative class. The red points have positive mean z-scores in both dimensions. Geometrically, ZSRI learns a half-open hyperrectangle that encloses the positive class. Depending on the thresholding strategy, the hyperrectangle may include or exclude outliers. In strategy (a), all positive points are included, while in (b), the outlier close to negative samples is excluded. Specifically, strategy (a) adds a safety buffer $\alpha > 0$ to enlarge the classification region.

2 Related Work

Our proposed rule induction algorithms intersect several established subfields of machine learning, including rule-based classification, mathematical optimization, and hyperbox classification.

2.1 Rule-Based Classification and Optimization

Rule-based classifiers map feature conditions to class labels via human-readable logical statements. Classic algorithms like RIPPER (Cohen, 1995) and CN2 (Clark & Niblett, 1989) employ a "divide-and-conquer" heuristic to sequentially select rules based on information-theoretic criteria. Modern approaches frame rule learning as mathematical optimization, using branch-and-bound optimization (CORELS; Angelino et al., 2017), MCMC sampling (Bayesian Rule Lists; Letham et al., 2015), ensemble rule extraction (RuleFit; Friedman & Popescu, 2008), or integer programming. For instance, Dash et al. (2018) formulate the search for optimal conjunctive and disjunctive rules as an integer linear program (ILP), solving it via linear programming relaxation and column generation. While accurate and established, these methods require either solving computationally demanding combinatorial optimization problems using iterative solvers or learning complex ensembles. In contrast, our proposed ZSRI algorithm induces rules directly in a single pass, avoiding iterative solvers entirely.

2.2 Hyperbox Classification

By constructing one-sided threshold rules (e.g., $x_j > \theta_j$), our proposed algorithm shares goals with subgroup discovery and "bump hunting". The Patient Rule Induction Method (PRIM; Friedman & Fisher, 1999) is a seminal bump hunting algorithm that seeks box-shaped regions (hyperrectangles) with high target density using a greedy top-down "peeling" and bottom-up "pasting" strategy. A related geometric perspective is adopted in hyperbox classification (e.g., Simpson, 1992), where decision boundaries are formed by axis-aligned orthogonal boxes. Both paradigms share our geometric formulation of axis-aligned hyperrectangles, but existing approaches rely on iterative greedy peeling and pasting, heuristic expansion-contraction cycles, or costly optimization. Our approach differs by directly learning rule thresholds from target-class distributions without the need to iterate multiple times through the dataset.

3 Problem Statement

We formalize the task of learning interpretable conjunctive rules for binary classification in this section. We present the optimization objective and analyze the computational complexity of the underlying problem.

3.1 Problem Formulation

We formulate the rule-based classification task as a supervised learning problem. Let $\mathcal{X} \subseteq \mathbb{R}^m$ denote the m-dimensional feature space of continuous attributes, and let $\mathcal{Y} = \{0, 1\}$ represent the binary label space, where 1 designates the target (positive) class and 0 designates the non-target (negative) class.

We are given a training dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$, where each sample $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_m^{(i)})^T \in \mathcal{X}$ is associated with a class label $y^{(i)} \in \mathcal{Y}$, and $x_j^{(i)}$ denotes the value of the j-th feature for the i-th sample. We partition the dataset into the subset of positive instances $\mathcal{D}^+ = \{(x^{(i)}, y^{(i)}) \in \mathcal{D} \mid y^{(i)} = 1\}$ and negative instances $\mathcal{D}^- = \{(x^{(i)}, y^{(i)}) \in \mathcal{D} \mid y^{(i)} = 0\}$, with sizes denoted by $n^+ = |\mathcal{D}^+|$ and $n^- = |\mathcal{D}^-|$, respectively.

Our objective is to learn a mapping function $f : \mathcal{X} \to \mathcal{Y}$ that accurately predicts the label y for a query sample $x = (x_1, x_2, \ldots, x_m)^T \in \mathcal{X}$. To ensure interpretability, we restrict the hypothesis space of f to the class of *conjunctive rules* with k terms, where k is a hyperparameter and $k \leq m$.

Definition 1 (Conjunctive Rule) *A conjunctive rule of length k is a conjunction of k threshold conditions over a selected subset of features $S \subseteq \{1, 2, \ldots, m\}$ with $|S| = k$. Formally:*

$$f(x) = \bigwedge_{j \in S} \mathbb{I}(x_j \bowtie_j \theta_j)$$

where $\mathbb{I}(\cdot)$ is the indicator function mapping true assertions to 1 and false assertions to 0, $\bowtie_j \in \{\geq, \leq\}$ is a comparison operator, and $\theta_j \in \mathbb{R}$ is a threshold value for feature j. A query sample x is predicted as positive ($f(x) = 1$) if and only if all k conditions are satisfied, else $f(x) = 0$.

Geometrically, a conjunctive rule defines a half-space in the feature space. We illustrate this concept in Figure 1 for a two-dimensional feature space (x_1, x_2) with $k = 2$ selected features. Here, the target positive class (shown in red) is concentrated in the upper-right region of the space, forming an orthant corner that classifies the positive class. In this specific example, the conjunctive rule to enclose the positive instances is given by $x_1 \geq \theta_1 \wedge x_2 \geq \theta_2$ (represented by the shaded region). For k selected features, the rule region lies inside one of the 2^k orthants.

The goal of our rule classifier is to determine the optimal thresholds θ_1 and θ_2 that cover the maximum number of positive samples while minimizing the inclusion of negative samples (shown in blue).

3.2 Optimization Objective

Mathematically, the learning task is to identify the optimal subset of features S^*, the comparison operators $\{\bowtie_j\}_{j \in S^*}$, and the thresholds $\{\theta_j\}_{j \in S^*}$ that maximize classification performance (e.g., accuracy, F1-score, Recall, Precision, etc.) on the true data distribution:

$$\max_{S, \{\bowtie_j\}, \{\theta_j\}} \mathbb{E}_{(x,y) \sim \mathcal{P}} \left[\mathcal{M}(f(x), y) \right] \quad \text{s.t.} \quad |S| = k \tag{1}$$

where $\mathcal{M}$ is an arbitrary classification metric and $\mathcal{P}$ is the joint data distribution.

Finding the globally optimal conjunctive rule for Equation 1 over a finite training dataset $\mathcal{D}$ of size n is NP-hard. Specifically, a brute-force search to find the optimal subset $S^* \subseteq \{1, 2, \ldots, m\}$ of size k out of m features has $\binom{m}{k}$ combinations. For each selected feature, there are 2 possible comparison operators, resulting in 2^k combinations for k features. Furthermore, along each feature axis, there are at most $n + 1$ thresholds that yield distinct binary splits on the n training samples, yielding $(n + 1)^k$ threshold combinations. Evaluating each candidate rule on the n samples requires $O(kn)$ operations, leading to a total combinatorial complexity of $O\left(\binom{m}{k} 2^k (n + 1)^k \cdot kn\right) \approx O(m^k (2n)^k \cdot kn)$. Because of this exponential dependency on the rule size k, finding the global optimum is computationally intractable for larger values of k and m. While conventional greedy heuristics (e.g., ID3 Quinlan (1986) and C4.5 Quinlan (1993) for decision trees) reduce the search space, they remain computationally intensive.

4 Z-SCORE RULE INDUCTION (ZSRI)

In this section, we present the *Z-Score Rule Induction (ZSRI)* algorithm for binary classification. ZSRI constructs an interpretable classifier directly in a single pass using z-scores of the features from the target class samples. We detail the algorithm and prove its computational runtime complexity in the following.

4.1 ALGORITHM DESCRIPTION

The objective of ZSRI is to construct a conjunction of k threshold rules over a subset of features $S \subseteq \{1, \ldots, m\}$ that accurately identifies the positive class ($y = 1$) for any query sample $x = (x_1, \ldots, x_m)^T$.

The ZSRI algorithm induces k conjunctive rules in a single pass through three steps:

1. **Feature Standardization:** We compute the z-score for each sample $x^{(i)} \in \mathcal{X}$ across all features $z(x_j^{(i)}) = \frac{x_j^{(i)} - \mu_j}{\sigma_j}$, where μ_j and σ_j are the empirical mean and standard deviation of the j-th feature.
2. **Positive Class Shift Calculation:** Next, we compute the average z-score over the n^+ positive samples for each feature: $\bar{z}_j^+ = \frac{1}{n^+} \sum_{i:y^{(i)}=1} z(x_j^{(i)})$.
3. **Feature Importance Ranking:** Features are ranked in descending order of $|\bar{z}_j^+|$, and we select the top k features with the largest distributional shifts, i.e., $S = \{\pi(1), \ldots, \pi(k)\}$, where π is a permutation of $\{1, \ldots, m\}$ such that $|\bar{z}_{\pi(1)}^+| \geq |\bar{z}_{\pi(2)}^+| \geq \cdots \geq |\bar{z}_{\pi(m)}^+|$.
4. **Rule Induction:** For each of the top k features, we infer the direction (operator $\bowtie_j$) and the decision threshold θ_j. The direction is determined by the sign of the positive class mean z-score $\bar{z}_j^+$:
   - If $\bar{z}_j^+ > 0$, feature values are shifted higher in the positive class, yielding $x_j \geq \theta_j$.
   - If $\bar{z}_j^+ \leq 0$, feature values are shifted lower in the positive class, yielding $x_j \leq \theta_j$.

   For determining the threshold θ_j, let $X_j^+ = \{x_j^{(i)} \mid y^{(i)} = 1\}$ denote the values of feature j across the positive class instances. We present two strategies that the user can choose from depending on the desired trade-off between recall and precision. The two strategies are also illustrated geometrically in Figure 1:

   (a) **Extremum-based (Optimized for Recall):** The threshold can be set using the extreme values (minimum or maximum) of X_j^+:

   $$\theta_j = \begin{cases} \min(X_j^+) - \alpha \cdot \sigma_j^+, & \text{if } \bar{z}_j^+ > 0 \\ \max(X_j^+) + \alpha \cdot \sigma_j^+, & \text{if } \bar{z}_j^+ \leq 0 \end{cases}$$

   where σ_j^+ is the standard deviation of feature j over the positive class samples, and $\alpha \geq 0$ is a user-defined scaling parameter. For every $\alpha \geq 0$, all positive training instances satisfy every individual rule condition as depicted in Figure 1(a), i.e., the recall value is 100%. For $\alpha = 0$, the rule boundary is tight (similar to a *minimum bounding rectangle* that is half-open). Setting $\alpha > 0$ widens the boundary to potentially include more positive samples from the test set.

   (b) **Quantile-based (Optimized for Precision):** The threshold is defined by a user-specified quantile parameter $q \in (0, 0.5)$ over the positive class:

   $$\theta_j = \begin{cases} \text{Quantile}(X_j^+, q), & \text{if } \bar{z}_j^+ > 0 \\ \text{Quantile}(X_j^+, 1-q), & \text{if } \bar{z}_j^+ \leq 0, \end{cases}$$

   where $\text{Quantile}(X, q)$ denotes the q-th quantile of the data X. As illustrated in Figure 1(b), setting $q \approx 0.1$ ensures the rule region covers approximately 90% of the positive class samples along each selected feature dimension, filtering out marginal outliers to tighten the decision boundary for higher precision.

Theorem 1 (ZSRI Training Complexity) *Let $\mathcal{D}$ be a training set of size n with m continuous features, and let $n^+ = |\mathcal{D}^+| \leq n$ denote the number of positive instances. The computational complexity of training the Z-Score Rule Induction (ZSRI) algorithm with k rules is $O(n \cdot m)$ for extremum-based thresholding and $O(n \cdot m + k \cdot n^+ \log n^+)$ for quantile-based thresholding.*

Proof: The complexity of each step is as follows:

1. *Feature Standardization:* Computing the mean and standard deviation for each of the m features across all n samples requires $O(n \cdot m)$ operations. Computing the z-scores for all samples also takes $O(n \cdot m)$ time.
2. *Positive Class Filtering:* Filtering the positive samples takes $O(n)$ time. Computing the average z-score $\bar{z}_j^+$ over the n^+ positive instances for all m features requires $O(n^+ \cdot m)$ additions and divisions.
3. *Feature Importance Ranking:* Sorting the m features by their absolute average z-score requires $O(m \log m)$ comparisons.
4. *Rule Induction:* For the top k selected features, setting decision thresholds depends on the chosen strategy: the extremum-based strategy evaluates extreme values and positive dispersion σ_j^+ over the n^+ positive instances in $O(n^+)$ time per feature (taking $O(k \cdot n^+)$ in total), whereas the quantile-based strategy requires sorting the n^+ positive instances, taking $O(n^+ \log n^+)$ per feature (taking $O(k \cdot n^+ \log n^+)$ for k features).

Combining steps 1, 2, and 3, the computational complexity is $O(n \cdot m + n^+ \cdot m + m \log m)$. Since $n^+ \leq n$, $k \leq m$, positive class filtering $O(n^+ \cdot m)$ and feature sorting $O(m \log m)$ are dominated by the feature standardization.

Depending on the threshold strategy in step 4, the added complexity is $O(k \cdot n^+)$ for extremum-based thresholding and $O(k \cdot n^+ \log n^+)$ for quantile-based thresholding. Thus, the overall training runtime is bounded by $O(n \cdot m)$ for extremum-based thresholding and $O(n \cdot m + k \cdot n^+ \log n^+)$ for quantile-based thresholding. ■

4.2 Probability Smoothing

While ZSRI offers interpretability, the hard discrete decision $f(x) \in \{0, 1\}$ abruptly assigns zero probability to near-miss query samples. We could simply use logistic regression to get probabilistic outcomes but this makes the whole idea of getting rules obsolete. Instead, we derive posterior probabilities $\hat{P}(Y = 1 \mid x) \in [0, 1]$ using the rules from ZSRI under the condition that $\hat{P}(Y = 1 \mid x) = 1$ if and only if $f(x) = 1$. We denote this region where all k conditions are satisfied as:

$$\mathcal{R} = \{x \in \mathcal{X} \mid \forall j \in S,\ x_j \bowtie_j \theta_j\} = \{x \in \mathcal{X} \mid f(x) = 1\}.$$

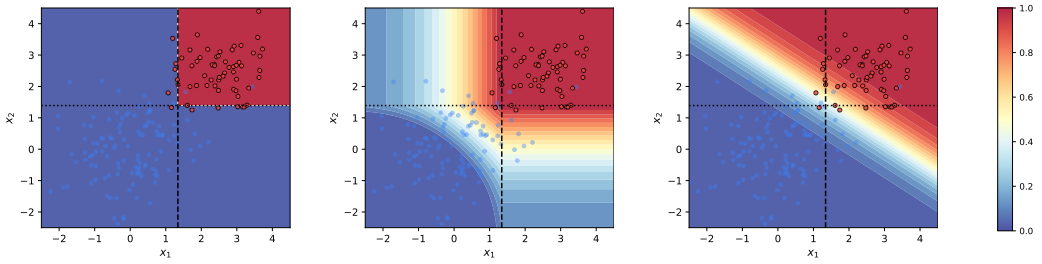
For negative predictions $x \notin \mathcal{R}$, i.e., $f(x) = 0$, we assign smoothed probabilities $\hat{P}(Y = 1 \mid x) \in [0, 1)$ based on which rules are violated and to what extent they are violated.

Our smoothing method works as follows: We are given the k selected features S, comparison operators $\bowtie_j$, and thresholds θ_j. For each rule $x_j \bowtie_j \theta_j$ that is not satisfied by query sample x, a penalty is added. The penalties altogether are summed up, normalized, and subtracted from 1 to obtain $\hat{P}(Y = 1 \mid x)$. A query sample x that satisfies all rules has no penalty and hence $\hat{P}(Y = 1 \mid x) = 1$. To define these penalties in a scale-invariant and probabilistic manner, we evaluate feature violations using the standard normal cumulative distribution function (CDF) Φ. For each feature $j \in S$, we map the query sample value x_j to its standardized z-score $z(x_j) = \frac{x_j - \mu_j}{\sigma_j}$ and the decision threshold θ_j to $z(\theta_j) = \frac{\theta_j - \mu_j}{\sigma_j}$.

Theorem 2 (Standardized Condition Invariance) *Any rule condition $x_j \bowtie_j \theta_j$ is preserved in standardized z-score coordinates:*

$$x_j \bowtie_j \theta_j \iff z(x_j) \bowtie_j z(\theta_j).$$

Proof: Since standardization is a strictly monotonic transformation ($\sigma_j > 0$), the inequalities are preserved. ■



(a) ZSRI (No Smoothing) (b) ZSRI (Smoothing) (c) Logistic Regression

Figure 2: Predicted positive probability fields $\hat{P}(Y = 1 \mid x)$ in a 2D feature subspace defined by rules $x_1 \geq \theta_1$ and $x_2 \geq \theta_2$: (a) ZSRI rules without smoothing, (b) ZSRI rules with our smoothing method, and (c) Logistic Regression.

Definition 2 (Single Rule Penalty) *Let $s_j = \operatorname{sgn}(\bar{z}_j^+) \in \{+1, -1\}$ denote the direction sign of feature $j \in S$, defined as*

$$s_j = \begin{cases} +1, & \bar{z}_j^+ > 0, \\ -1, & \bar{z}_j^+ \leq 0, \end{cases}$$

where $s_j = +1$ corresponds to the condition $x_j \geq \theta_j$, and $s_j = -1$ corresponds to the condition $x_j \leq \theta_j$. The penalty $\delta_j^+(x)$ for query sample x along feature j is defined as follows:

$$\delta_j^+(x) = \max\left(0,\ s_j \cdot (\Phi(z(\theta_j)) - \Phi(z(x_j)))\right).$$

The penalty makes use of the difference of the CDF values $\Phi(z(\cdot))$ of the z-scores $z(\theta_j)$ and $z(x_j)$. The sign s_j is to rectify $s_j(\Phi(z(\theta_j)) - \Phi(z(x_j))) \leq 0$ to $\delta_j^+(x) = 0$, if the condition $x_j \bowtie_j \theta_j$ is satisfied. When condition j is violated, $\delta_j^+(x) > 0$ measures the probability mass by which x_j falls beyond the decision threshold $z(\theta_j)$.

Definition 3 (Total Maximum Penalty) *The maximum penalty M_j along feature dimension $j \in S$ occurs in the extreme limit of feature violation. Depending on s_j, it is defined as the CDF difference from the threshold $z(\theta_j)$ to negative or positive infinity:*

$$M_j = \begin{cases} \Phi(z(\theta_j)) - \Phi(-\infty) = \Phi(z(\theta_j)), & s_j = +1, \\ \Phi(+\infty) - \Phi(z(\theta_j)) = 1 - \Phi(z(\theta_j)) = \Phi(-z(\theta_j)), & s_j = -1, \end{cases}$$

which is compactly written as $M_j = \Phi(s_j \cdot z(\theta_j))$. Across all $k = |S|$ selected rule features, the total maximum penalty is defined as the sum of maximum penalties across features: $\sum_{j \in S} M_j$.

Definition 4 (Smooth Probability Estimator) *Normalizing the aggregated penalty across all violated features by the total maximum penalty yields the normalized deficit $\bar{D}(x) \in [0, 1]$:*

$$\bar{D}(x) = \frac{\sum_{j \in S} \delta_j^+(x)}{\sum_{j \in S} M_j} = \frac{\sum_{j \in S} \max\left(0,\ s_j \cdot (\Phi(z(\theta_j)) - \Phi(z(x_j)))\right)}{\sum_{j \in S} \Phi(s_j \cdot z(\theta_j))}.$$

The smooth posterior probability is then obtained by subtracting this normalized penalty from 1:

$$\hat{P}(Y = 1 \mid x) = 1.0 - \bar{D}(x).$$

Theorem 3 (Rule Boundary Preservation) *For any query sample $x \in \mathcal{X}$, the smooth probability estimator satisfies $\hat{P}(Y = 1 \mid x) = 1.0$ if and only if $x \in \mathcal{R}$. Furthermore, for any sample $x \notin \mathcal{R}$ that strictly violates at least one rule threshold, $\hat{P}(Y = 1 \mid x) < 1.0$.*

Proof: For any sample $x \in \mathcal{R}$, all rule conditions $x_j \bowtie_j \theta_j$ are satisfied for all $j \in S$, which yields $\delta_j^+(x) = 0$ for all $j \in S$. Plugging these into $\bar{D}(x)$ gives $\bar{D}(x) = \frac{0}{\sum_{j \in S} M_j} = 0$, and thus $\hat{P}(Y = 1 \mid x) = 1.0 - 0 = 1.0$. Conversely, for any sample $x \notin \mathcal{R}$, at least one condition $j^* \in S$

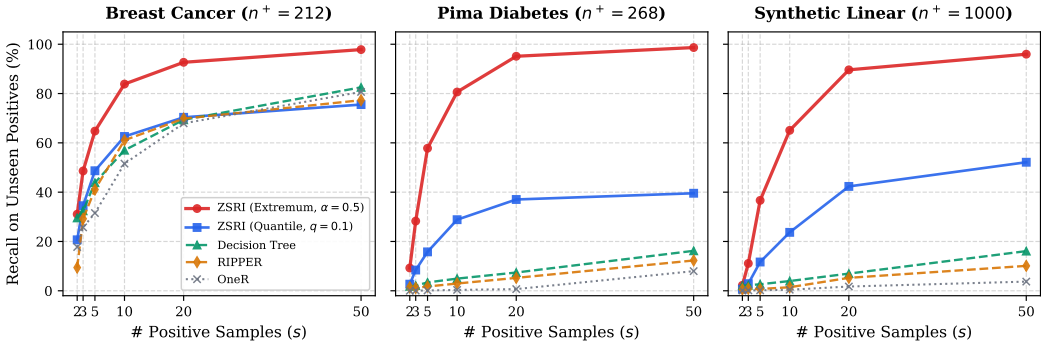


Figure 3: Few-shot positive rule discovery performance: Retrieval Recall on unseen positive instances across varying numbers of seed positive samples $s \in \{2, 3, 5, 10, 20, 50\}$ for Wisconsin Breast Cancer, Pima Indians Diabetes, and Synthetic Linear benchmarks.

is violated, yielding $\delta_{j^*}^+(x) > 0$. Since $M_j > 0$, this gives $\bar{D}(x) = \frac{\sum_{j \in S} \delta_j^+(x)}{\sum_{j \in S} M_j} > 0$, and thus $\hat{P}(Y = 1 \mid x) = 1.0 - \bar{D}(x) < 1.0$. ■

As illustrated in Figure 2, the discrete decision rule (Figure 2a) creates a step function where any query sample close but outside the acceptance region $\mathcal{R}$ is assigned zero probability. In contrast, with our smoothing approach (Figure 2b), samples near the boundary retain high probabilities while preserving full certainty ($\hat{P} = 1$) inside $\mathcal{R}$. In a medical context, the smoothing approach creates a risk landscape of patients with a high probability of disease. Conversely, Logistic Regression (Figure 2c) fits a linear decision boundary across the space, failing to enforce the axis-aligned rule boundaries or guarantee full certainty inside $\mathcal{R}$.

5 Experiments

To evaluate the empirical performance, few-shot capability, and scalability of the proposed Z-Score Rule Induction (ZSRI) algorithm, we formulate three research questions: **RQ1 (Few-Shot Positive Rule Discovery):** Can ZSRI induce effective bounding rules from only a tiny seed of positive examples ($s \ll n^+$) to retrieve unseen positive instances from unlabeled populations? **RQ2 (Predictive Performance):** How does the predictive performance of ZSRI compare to Decision Trees and alternative rule-based learners on standard benchmarks? **RQ3 (Computational Scalability):** How does the training execution runtime of ZSRI scale compared to recursive partition-based Decision Trees as dataset size increases?

5.1 RQ1: Few-Shot Positive Rule Discovery and Retrieval

In many domains, confirmed positive cases are scarce and expensive to obtain (e.g., rare medical conditions, biomarkers, etc.), while background/unlabeled instances are abundant. We investigate whether ZSRI can discover an accurate rule from only a tiny seed of s positive samples ($s \ll n^+$) and successfully retrieve the remaining positive instances in the population without retrieving too many false positives.

5.1.1 Experimental Protocol

For each dataset, we randomly sample a small seed set of $s \in \{2, 3, 5, 10, 20, 50\}$ positive instances from $\mathcal{D}^+$. The models are trained on these s positive examples together with the background dataset $\mathcal{D}^-$. We then evaluate the induced rule on the *unseen* positive instances ($\mathcal{D}^+ \setminus \mathcal{D}^+_{\text{seed}}$) alongside all negative instances. We report the mean across 30 independent random seed draws for Recall (Sensitivity), Precision, Specificity (True Negative Rate), and F1-score.

Table 1: Predictive benchmark comparison on standard 70-30 train-test partitions (unabridged results with Accuracy and runtimes in Appendix B).

	Breast Cancer			Synthetic Linear			Pima Diabetes		
Model	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
ZSRI (Quantile)	**1.000**	0.787	0.881	0.787	0.623	0.696	0.620	0.550	0.583
ZSRI (Extremum)	0.856	**0.991**	0.918	0.519	**1.000**	0.683	0.346	**1.000**	0.514
Decision Tree	0.963	0.963	0.963	0.803	0.714	0.756	0.667	0.550	**0.603**
OneR	0.933	0.898	0.915	0.739	0.627	0.678	**0.692**	0.338	0.454
SkopeRules	0.979	0.852	0.911	0.000	0.000	0.000	0.000	0.000	0.000
BoostedRules	0.973	**0.991**	**0.982**	0.741	0.799	**0.769**	0.673	0.463	0.548
RIPPER	0.936	0.954	0.945	**0.841**	0.360	0.505	0.619	0.487	0.545

5.1.2 Results and Discussion

The retrieval curves across seed budgets are illustrated in Figure 3, with the comprehensive numerical grid across all metrics reported in Appendix A (Table 2). The results highlight several fundamental advantages of ZSRI in the few-shot regime:

Rapid Positive Retrieval with Ultra-Small Seeds: On Breast Cancer, using only $s = 3$ positive examples (out of $n^+ = 212$), ZSRI (Extremum, $\alpha = 0.5$) retrieves 48.6% of all unseen malignant cases with 98.9% precision and 98.7% specificity (F1: 0.615). With $s = 5$, ZSRI captures 64.8% of unseen positives (98.4% precision, 98.0% specificity), and with $s = 10$, it retrieves 83.9% of positives with an F1-score of 0.894.

Overcoming Tree and Sequential Covering Collapse: Standard Decision Trees and RIPPER suffer severe retrieval collapse in few-shot regimes. On Pima Diabetes with $s = 5$, Decision Trees achieve only 3.4% Recall, OneR achieves 0.1%, and RIPPER achieves 1.7%. Because these algorithms optimize symmetric impurity splits, extreme class imbalance forces them to prune branches or isolate only the training points, resulting in near-total failure to retrieve unseen positives. In contrast, ZSRI (Extremum, $\alpha = 0.5$) achieves 57.8% Recall with $s = 5$ and 80.6% Recall with $s = 10$.

Controlled False Positives (High Specificity): Crucially, ZSRI does not achieve high recall by trivially classifying all samples as positive. Across all seed sizes $s \in \{3, 5, 10, 20, 50\}$, ZSRI (Quantile) maintains specificities between 85.6% and 99.7% and precisions between 54.3% and 99.6%. ZSRI (Extremum) similarly maintains specificities above 80% in low-noise regimes while expanding the boundary to guarantee high sensitivity.

These findings confirm that ZSRI is well-suited for positive-unlabeled retrieval and few-shot rule induction, allowing practitioners to extract human-readable bounding rules from a handful of exemplar cases to retrieve target instances across large candidate datasets.

5.2 RQ2: Predictive Performance on Standard Benchmarks

We next evaluate ZSRI on fully supervised benchmarks using a 70-30 train-test partition across three continuous datasets: Breast Cancer Wisconsin ($N = 569$, $d = 30$, $n^+ = 212$), Synthetic Linear ($N = 2,000$, $d = 20$, $n^+ = 1,000$), and Pima Indians Diabetes ($N = 768$, $d = 8$, $n^+ = 268$). We compare against standard rule-based and tree baselines: (1) ZSRI (Quantile, $q = 0.1$); (2) ZSRI (Extremum, $\alpha = 0.5$); (3) Decision Tree (depth 5); (4) OneR; (5) SkopeRules (10 estimators, depth 3); (6) BoostedRules (5 estimators); and (7) RIPPER.

Table 1 summarizes the classification performance. We observe clear, predictable trade-offs between the two ZSRI threshold strategies: **ZSRI (Quantile)** prioritizes precision by defining a tight hyperrectangle around the positive core. On Breast Cancer, it achieves a perfect Precision of 1.000 (Recall 0.787, F1 0.881), outperforming SkopeRules (0.979) and Decision Tree (0.963) in precision. On Synthetic Linear, it achieves 0.787 Precision and 0.696 F1. **ZSRI (Extremum)** maximizes coverage and recall by buffering positive dispersion with a standard deviation margin. On Breast Cancer, it achieves 0.991 Recall and 0.918 F1, outperforming OneR (0.915) and SkopeRules (0.911). On both Synthetic Linear and Pima Diabetes, it achieves a perfect Recall of 1.000, ensuring zero missed

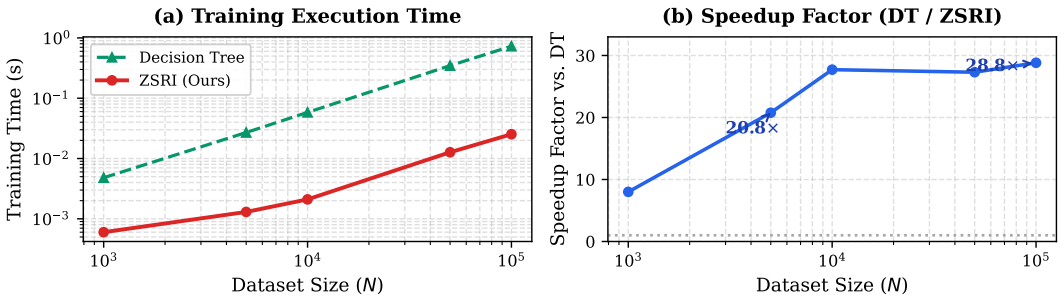


Figure 4: Computational scalability: (a) Wall-clock training execution time across dataset sizes $N \in [10^3, 10^5]$ (log-log scale); (b) Speedup factor achieved by ZSRI over Decision Trees ($28.8\times$ speedup at $N = 100,000$).

positive cases in critical diagnostic settings. On Synthetic Linear and Pima Diabetes, SkopeRules fails entirely to predict positive labels (F1 of 0.000) due to overly aggressive pruning, whereas both ZSRI variants remain robust and competitive with Decision Trees while utilizing only 5 simple, interpretable conjunctive rules.

5.3 RQ3: Computational Efficiency and Scalability

To evaluate computational scalability, we measured the training execution time of ZSRI and the Decision Tree baseline on synthetic datasets of increasing size $N \in [1,000, 100,000]$ ($m = 20$ features, 10 informative).

As shown in Figure 4 (and detailed in Appendix C, Table 4), ZSRI scales strictly linearly with sample size $O(n \cdot m)$. While Decision Trees require recursive sorting and impurity evaluations across candidate cut points ($O(m \cdot n \log n)$ per tree level), ZSRI computes positive-class moments and filter quantiles in a single pass. Consequently, ZSRI delivers an execution speedup of $20.6\times$ at $N = 5,000$ and up to $28.8\times$ at $N = 100,000$ (0.025 s vs. 0.729 s for Decision Tree). Moreover, compared to iterative sequential covering algorithms such as RIPPER, which requires 0.14 s to 0.47 s on small datasets ($N < 1,000$), ZSRI is over $100\times$ faster.

6 Conclusion

In this paper, we introduced the Z-Score Rule Induction (ZSRI) algorithm, a lightweight, interpretable method for rule discovery and binary classification. By identifying feature axes where the positive class significantly deviates from the background population and defining robust quantile- or extremum-based bounds, ZSRI constructs simple conjunctions of rules directly in a single pass without recursive partitioning. Empirical evaluations across medical and synthetic benchmarks demonstrate that ZSRI achieves competitive predictive performance while executing up to $28\times$ faster than standard decision trees. Crucially, in few-shot learning with sparse labeled positives, ZSRI reliably induces rules from as few as $\{3, 5, 10\}$ positive examples to retrieve unseen positive cases. To accommodate more practical applications, we also introduce a smoothing method for the induced rules to return probabilities. Near-miss samples get assigned a high probability instead of a negative class label. This method comes with theoretical guarantees. Future work includes extending ZSRI to multi-class classification.

AI Use Disclosure

In this work, generative AI tools were used to assist coding our algorithm (ZSRI, smoothing method, and experiments), literature search, and polishing text for readability. Generative AI was not used for formulating the theoretical framework, mathematical theorems, proofs, or interpreting the results. However, we did ask it to check our reasoning in those places. We developed the main ideas, the overall algorithm, and the main conclusions ourselves. All AI-assisted code and citations were manually verified. The authors take full responsibility for the final content of this work.

REFERENCES

Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 35–44, 2017.

Peter Clark and Tim Niblett. The cn2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989. doi: 10.1007/BF00116835.

William W. Cohen. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML-95)*, pp. 115–123. Morgan Kaufmann, 1995.

Sanjeev Dash, Oktay Günlük, and Dennis Wei. Boolean decision rules via column generation. *Advances in Neural Information Processing Systems*, 31:4660–4670, 2018.

Jerome H. Friedman and Nicholas I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123–143, 1999.

Jerome H. Friedman and Bogdan E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.

Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

Patrick K. Simpson. Fuzzy min-max neural networks—Part 1: Classification. *IEEE Transactions on Neural Networks*, 3(5):776–786, 1992. doi: 10.1109/72.159066.

Table 2: Few-Shot Positive Rule Discovery: Retrieval performance on unseen test data when training on only s positive instances ($s \ll n^+$) with full/unlabeled background.

		Breast Cancer Wisconsin ($n^+ = 212$)					**Pima Indians Diabetes** ($n^+ = 268$)					**Synthetic Linear** ($n^+ = 1000$)				
Model	**Metric**	$s = 3$	$s = 5$	$s = 10$	$s = 20$	$s = 50$	$s = 3$	$s = 5$	$s = 10$	$s = 20$	$s = 50$	$s = 3$	$s = 5$	$s = 10$	$s = 20$	$s = 50$
ZSRI (Extremum, $\alpha = 0.5$)	Recall	0.486	0.648	0.839	0.927	0.978	0.283	0.578	0.806	0.951	0.987	0.111	0.367	0.651	0.896	0.960
	Prec.	0.989	0.984	0.967	0.941	0.893	0.526	0.512	0.419	0.372	0.315	0.724	0.663	0.618	0.580	0.541
	F1	0.615	0.763	0.894	0.932	0.933	0.316	0.511	0.541	0.534	0.477	0.175	0.440	0.620	0.702	0.692
	Spec.	0.987	0.980	0.950	0.905	0.828	0.837	0.666	0.402	0.194	0.064	0.956	0.796	0.587	0.352	0.220
ZSRI (Quantile, $q = 0.1$)	Recall	0.345	0.487	0.626	0.704	0.755	0.084	0.158	0.288	0.370	0.396	0.029	0.117	0.237	0.423	0.521
	Prec.	0.996	0.994	0.991	0.990	0.988	0.624	0.678	0.601	0.571	0.543	0.841	0.804	0.783	0.766	0.733
	F1	0.479	0.629	0.760	0.821	0.855	0.138	0.247	0.375	0.439	0.448	0.054	0.192	0.352	0.539	0.604
	Spec.	0.997	0.994	0.991	0.989	0.987	0.969	0.956	0.896	0.858	0.856	0.995	0.969	0.930	0.867	0.812
Decision Tree	Recall	0.324	0.438	0.570	0.694	0.825	0.021	0.034	0.049	0.074	0.162	0.022	0.026	0.040	0.070	0.161
	Prec.	1.000	1.000	1.000	1.000	0.996	0.967	0.987	0.981	0.934	0.904	1.000	0.998	0.999	0.997	0.992
	F1	0.480	0.601	0.721	0.817	0.902	0.040	0.065	0.092	0.135	0.270	0.044	0.051	0.075	0.129	0.273
	Spec.	1.000	1.000	1.000	1.000	0.995	1.000	1.000	0.999	0.998	0.992	1.000	1.000	1.000	1.000	0.998
RIPPER	Recall	0.292	0.411	0.611	0.698	0.773	0.011	0.017	0.029	0.052	0.123	0.009	0.007	0.014	0.052	0.101
	Prec.	0.981	0.992	0.990	0.986	0.978	0.699	0.678	0.646	0.809	0.779	0.735	0.776	0.884	0.948	0.971
	F1	0.404	0.557	0.741	0.811	0.861	0.022	0.033	0.055	0.096	0.200	0.016	0.014	0.028	0.097	0.181
	Spec.	0.995	0.994	0.988	0.983	0.974	0.998	0.997	0.996	0.994	0.984	0.999	0.999	0.998	0.998	0.997
OneR	Recall	0.257	0.315	0.516	0.679	0.807	0.001	0.001	0.003	0.007	0.080	0.002	0.002	0.004	0.017	0.037
	Prec.	0.966	0.998	0.992	0.983	0.972	0.067	0.133	0.313	0.303	0.509	0.267	0.433	0.524	0.824	0.846
	F1	0.398	0.460	0.667	0.800	0.881	0.001	0.003	0.007	0.013	0.124	0.004	0.004	0.009	0.032	0.068
	Spec.	1.000	0.998	0.992	0.980	0.966	1.000	1.000	1.000	0.999	0.992	1.000	1.000	1.000	1.000	0.999

A Exhaustive Few-Shot Positive Rule Discovery Grid

Table 2 presents the comprehensive numerical results for the few-shot positive rule discovery and target instance retrieval benchmark described in Section 5.1. We report the mean across 30 independent random seed draws for Recall (Sensitivity on unseen positives $\mathcal{D}^+ \setminus \mathcal{D}^+_{\text{seed}}$), Precision, Specificity (True Negative Rate on background samples), and F1-score across seed budgets $s \in \{3, 5, 10, 20, 50\}$.

The numerical breakdown demonstrates that while standard Decision Trees and RIPPER exhibit near-zero recall under severe positive sparsity ($s \leq 5$) due to recursive impurity pruning, ZSRI (Extremum) guarantees rapid positive recovery (83.9% on Breast Cancer and 80.6% on Pima Diabetes at $s = 10$) while maintaining strong specificity (95.0% on Breast Cancer). Meanwhile, ZSRI (Quantile) maintains ultra-high precision ($> 98\%$ on Breast Cancer and $> 73\%$ on Synthetic Linear) across all seed regimes.

B Extended Full-Dataset Benchmark Results

Table 3 provides the unabridged benchmark results across all candidate algorithms, including the Greedy Rule Classifier baseline and per-model training wall-clock runtimes on a standard 70-30 train-test partition.

C Computational Scalability Measurements

Table 4 reports the exact wall-clock training execution times (averaged over multiple repetitions) as the dataset size N scales from $1,000$ to $100,000$ synthetic samples ($m = 20$ features, 10 informative), alongside the empirical speedup factor achieved by ZSRI over standard Decision Trees.

D Rule Interpretability and Decision Tree Split Congruence

To evaluate whether ZSRI's single-pass filter selection recovers physically meaningful and coherent decision boundaries, we compare the logical conjunction rules induced by ZSRI on the Wisconsin Breast Cancer benchmark against the top split decisions of a standard Decision Tree (depth 3) and a sequential Greedy Rule Classifier.

Table 5 details the top-5 feature conditions, induced thresholds, and selection scores extracted by ZSRI and Greedy Rule Classifier, alongside the root and depth-1 split decisions of the Decision Tree.

We observe a striking congruence between the features selected by ZSRI and the top split decisions of the Decision Tree:

Table 3: Unabridged benchmark results comparing all rule-based classifiers and baseline algorithms across three evaluation datasets.

Dataset	Model	Accuracy	Precision	Recall (Sens.)	F1	Time (s)
Breast Cancer	ZSRI (Quantile)	0.865	1.000	0.787	0.881	0.001
Breast Cancer	ZSRI (Std)	0.889	0.856	0.991	0.918	0.000
Breast Cancer	Greedy Rule Classifier	0.889	1.000	0.824	0.904	0.003
Breast Cancer	Decision Tree	0.953	0.963	0.963	0.963	0.004
Breast Cancer	OneR	0.895	0.933	0.898	0.915	0.049
Breast Cancer	SkopeRules	0.895	0.979	0.852	0.911	0.089
Breast Cancer	BoostedRules	0.977	0.973	0.991	0.982	0.008
Breast Cancer	RIPPER	0.930	0.936	0.954	0.945	0.403
Synthetic Linear	ZSRI (Quantile)	0.720	0.787	0.623	0.696	0.001
Synthetic Linear	ZSRI (Std)	0.523	0.519	1.000	0.683	0.001
Synthetic Linear	Greedy Rule Classifier	0.720	0.787	0.623	0.696	0.003
Synthetic Linear	Decision Tree	0.763	0.803	0.714	0.756	0.010
Synthetic Linear	OneR	0.695	0.739	0.627	0.678	0.052
Synthetic Linear	SkopeRules	0.487	0.000	0.000	0.000	0.114
Synthetic Linear	BoostedRules	0.753	0.741	0.799	0.769	0.016
Synthetic Linear	RIPPER	0.637	0.841	0.360	0.505	0.475
Pima Diabetes	ZSRI (Quantile)	0.727	0.620	0.550	0.583	0.001
Pima Diabetes	ZSRI (Std)	0.346	0.346	1.000	0.514	0.000
Pima Diabetes	Greedy Rule Classifier	0.697	0.569	0.512	0.539	0.001
Pima Diabetes	Decision Tree	0.749	0.667	0.550	0.603	0.002
Pima Diabetes	OneR	0.719	0.692	0.338	0.454	0.017
Pima Diabetes	SkopeRules	0.654	0.000	0.000	0.000	0.064
Pima Diabetes	BoostedRules	0.736	0.673	0.463	0.548	0.005
Pima Diabetes	RIPPER	0.719	0.619	0.487	0.545	0.145

Table 4: Training execution times and speedup factors across increasing dataset sizes N.

Dataset Size (N)	**ZSRI (s)**	**Greedy RC (s)**	**Decision Tree (s)**	**Speedup (DT/ZSRI)**
1,000	0.0006	0.0023	0.0048	7.9×
5,000	0.0013	0.0046	0.0270	20.6×
10,000	0.0021	0.0068	0.0582	27.5×
50,000	0.0127	0.0273	0.3467	27.3×
100,000	0.0253	0.0524	0.7292	28.8×

- **Feature Overlap:** ZSRI selects `worst concave points`, `worst perimeter`, `mean concave points`, `worst radius`, and `mean perimeter`. The Decision Tree selects `worst radius` at the root split and `worst concave points` at depth 1. Both models identify this exact subset of size- and shape-related nuclear measurements as primary indicators of the positive class.
- **Threshold Similarity:** For `worst radius`, ZSRI selects a decision threshold of ≤ 16.050, which aligns closely with the Decision Tree's split boundary at ≤ 16.795. For `worst concave points`, ZSRI selects ≤ 0.122, aligning with the tree's split at ≤ 0.136. Similarly, for `worst perimeter`, ZSRI selects ≤ 104.500.

This alignment demonstrates that univariate z-score filter ranking successfully recovers the main decision boundaries identified by recursive combinatorial tree induction algorithms, but in a much more direct, interpretable conjunctive format that executes in a single pass.

Table 5: Detailed rule comparison between ZSRI, Greedy Rule Classifier, and Decision Tree split decisions on the Breast Cancer Wisconsin dataset.

Classifier	**Order / Split**	**Rule Condition**	**Score / Threshold**
ZSRI	Rule 1	`worst concave points` ≤ 0.122	-0.612 (z)
	Rule 2	`worst perimeter` ≤ 104.500	-0.603 (z)
	Rule 3	`mean concave points` ≤ 0.048	-0.598 (z)
	Rule 4	`worst radius` ≤ 16.050	-0.598 (z)
	Rule 5	`mean perimeter` ≤ 94.174	-0.572 (z)
Greedy RC	Rule 1	`worst perimeter` ≤ 104.500	–
	Rule 2	`worst radius` ≤ 16.050	–
	Rule 3	`worst area` ≤ 779.340	–
	Rule 4	`mean radius` ≤ 14.452	–
	Rule 5	`mean perimeter` ≤ 94.174	–
Decision Tree	Root Split	`worst radius` ≤ 16.795	–
	Depth 1 (Left)	`worst concave points` ≤ 0.136	–
	Depth 1 (Right)	`mean texture` ≤ 16.110	–