Knowledge and Information Systems (2026) 68:250
https://doi.org/10.1007/s10115-026-02872-5

RESEARCH

General OOD detection via model-aware and subspace-aware variable priority

Min Lu[1] · **Hemant Ishwaran**[1]

Received: 12 December 2025 / Revised: 5 August 2026 / Accepted: 10 August 2026

Abstract
Out-of-distribution (OOD) detection flags test inputs that depart from the data used to train a model. For structured tabular problems with regression or survival outcomes, existing methods remain limited because many OOD detectors are designed for classification or unstructured data. We introduce a tree based method that uses the rule structure of a supervised forest to determine the predictive subspace, the model-aware reference neighborhood, and the final OOD score in a single construction. The score compares each test input with forest selected reference cases only on prediction relevant variables, reducing signal dilution from nuisance coordinates. Across synthetic and real data benchmarks, the method performs especially well for subtle targeted feature shifts and changes in dependence. An esophageal cancer survival study further shows how OOD scores can reveal lymphadenectomy related shifts relevant to surgical guidelines.

Keywords Predictive subspace · Random forests · Survival analysis · Tabular supervised learning · Tree rules · Variable priority

1 Introduction

Most prediction models are designed to return predictions, not to indicate when a new case is poorly supported by the data. A new input may differ from the training population in a single influential covariate, in a subtle combination of variables, or in the dependence structure among covariates, yet the model may still return a numerical prediction, survival curve, or risk score that may appear routine even though the test case should be considered suspicious. In settings such as clinical medicine and medical informatics, where predictions may influence diagnosis, monitoring, or treatment, this can have serious consequences. *Out-of-distribution* (OOD) detection addresses this problem by flagging inputs that are poorly represented by the data used to train the model [1, 2]. Because the forms of distributional departure that arise after deployment are rarely known in advance, an OOD detector must often be learned from *in-distribution* (ID) data alone.

✉ Hemant Ishwaran
hishwaran@med.miami.edu

Min Lu
m.lu6@umiami.edu

[1] Division of Biostatistics, Miller School of Medicine, University of Miami, Miami, USA

We study OOD detection for supervised tabular prediction with continuous or time-to-event outcomes. Much of the OOD literature has been developed for image and text classification, where class probabilities, logits, and learned representations provide natural ingredients for OOD scoring. Regression and survival problems lack this class-based structure. When such problems are recorded in tabular form, however, the covariates are explicit, and the trained model can reveal how those covariates support prediction. The key idea in this paper is to use this model-derived structure to assess unusualness relative to the prediction task rather than relative to the covariate distribution alone. Specifically, we use the rule structure of a supervised random forest to identify coordinates that contribute to prediction and to isolate reference training cases that are locally relevant for a test input. The resulting OOD score compares the test input with those reference cases within the prediction-relevant covariate subspace.

To place this approach in the standard distribution-shift framework, write $\mathbb{P}_{X,Y}$ for the joint law of (X, Y) in the training population, with marginal law $\mathbb{P}_X$ and conditional law $\mathbb{P}_{Y|X}$. Let $\mathbb{P}^*_{X,Y}$ denote the corresponding test law, with marginal $\mathbb{P}^*_X$ and conditional law $\mathbb{P}^*_{Y|X}$. In the tabular settings considered here, $X = (X^{(1)}, \ldots, X^{(d)}) \in \mathbb{R}^d$ is the structured covariate vector used to predict the outcome Y. The test distribution is OOD relative to the training distribution whenever $\mathbb{P}^*_{X,Y} \neq \mathbb{P}_{X,Y}$. Such a difference may arise through covariate shift, in which $\mathbb{P}_X$ changes while $\mathbb{P}_{Y|X}$ remains fixed; conditional or concept shift, in which $\mathbb{P}_{Y|X}$ changes; or changes in both components [3–8].

This taxonomy describes how training and test laws can differ, but it does not by itself determine which differences should make an individual input suspicious for a supervised predictor. A change in $\mathbb{P}_X$ along nonpredictive coordinates can make a case atypical in the full covariate space while leaving it well supported for prediction [9]. Conversely, a localized change involving a small number of influential variables, or a change in their dependence structure, may be difficult to recognize in the full covariate space but important for prediction. A supervised OOD score should therefore reflect how variables enter the prediction rule rather than treating every detectable departure in $\mathbb{P}_X$ as equally consequential [10]. Our score implements this principle by comparing each input with forest-selected training cases in the covariate subspace used for prediction.

Although this approach is very general, we note that a limitation of our method, shared by other OOD input methods, is that it cannot detect shifts that leave the covariate law unchanged. For a new case, our score uses only the case covariates and the fixed forest learned from the ID covariates and outcomes; no outcomes from the test distribution are observed when the score is computed. Consequently, a pure concept shift, with $\mathbb{P}^*_{Y|X} \neq \mathbb{P}_{Y|X}$ but $\mathbb{P}^*_X = \mathbb{P}_X$, is OOD under the joint law but leaves no observable covariate signal for the detector. Detecting such a shift requires outcomes from the test distribution or additional assumptions, and is outside the scope of this paper.

1.1 Predictive subspace

We formalize the part of the covariate space relevant to prediction using a predictive subspace. Let $S_0 \subset \{1, \ldots, d\}$ denote a set of coordinates through which the training conditional law factors. For any $x \in \mathbb{R}^d$, write $x^{(S_0)} = (x^{(j)})_{j \in S_0}$. We suppose

$$\mathbb{P}(Y \in A \mid X = x) = \mathbb{P}(Y \in A \mid X^{(S_0)} = x^{(S_0)})$$

for every measurable set A and $\mathbb{P}_X$-almost every x. Thus, under the training distribution, the conditional law of Y depends on X only through $X^{(S_0)}$. We do not require S_0 to be unique

or minimal. Any set satisfying this condition provides a valid population target, although the purpose of the restriction is to remove coordinates that are unnecessary for prediction.

The set S_0 describes the population target of the subspace restriction, but it is unknown in practice. Our method uses variable priority from a supervised forest to construct an empirical signal set $\widehat{S}$ and then uses the same forest to determine the local reference set for a test input. Thus, S_0 motivates the subspace restriction, whereas $\widehat{S}$ is the learned subspace used by the score.

1.2 Approach

The proposed method rests on two properties. First, the OOD score depends on the trained supervised model, not only on the geometry or density of the covariates. Second, once the relevant training cases are identified, the final comparison uses only coordinates containing predictive information.

The terms *model-aware* and *subspace-aware* refer to these two operational properties. Fix a training procedure $\mathcal{P}$ and hold any internal algorithmic randomness fixed. Given training data $\mathcal{L} = \{(x_i, y_i)\}_{i=1}^n$, write $\hat{f} = \mathcal{P}(\mathcal{L})$ and $x_{1:n} = (x_1, \ldots, x_n)$.

Definition 1 *(Model-aware OOD score)* An OOD score $d(x^*; x_{1:n}, \hat{f})$ is *model-aware* if the trained supervised model $\hat{f}$ enters the construction of the score. Thus, holding x^* and $x_{1:n}$ fixed, changing the training outcomes can change the score through $\hat{f}$.

Definition 2 *(Subspace-aware OOD score)* Let $S \subset \{1, \ldots, d\}$ be a selected set of coordinates, and let $\mathcal{R}(x^*; x_{1:n}, \hat{f}) \subseteq \{x_1, \ldots, x_n\}$ denote the reference set used for test input x^*. An OOD score is *subspace-aware with respect to S* if, after the reference set has been formed, the final discrepancy depends on x^* and the reference points only through their coordinates in S.

Our construction satisfies both properties with a single learned object. We train a supervised forest on the ID data and use its decision rules in two connected ways. First, the rules identify the empirical signal set $\widehat{S}$. Second, for a test point x^*, we relax the rules containing x^* one selected coordinate at a time, rank the training points, and form a neighborhood $\mathcal{N}_K(x^*)$. This neighborhood is the reference set $\mathcal{R}(x^*; x_{1:n}, \hat{f})$ in Definition 2. The final OOD score averages the distance from x^* to this reference set, computed only on $\widehat{S}$,

$$d(x^*) = \frac{1}{K} \sum_{z \in \mathcal{N}_K(x^*)} D_{\widehat{S}}(x^*, z),$$

where $D_{\widehat{S}}$ denotes a distance using only the coordinates in $\widehat{S}$, with optional weights reflecting their relative importance. Thus, the score depends on $\hat{f}$ through both $\widehat{S}$ and $\mathcal{N}_K(x^*)$, satisfying Definition 1, while its final discrepancy uses only $\widehat{S}$, satisfying Definition 2.

1.3 Main contribution and evaluation

Our main contribution is an integrated model-aware and subspace-aware OOD score for tabular supervised learning. A single trained supervised forest identifies prediction-relevant variables, constructs a local reference set for each test input through learned rule regions, and computes a weighted distance on the selected variables. This construction differs from approaches that first reduce dimension or select features and then apply a standard distance, nearest-neighbor, or density-based anomaly score.

The empirical sections evaluate the construction in three ways. First, matched ablations separate the effects of variable priority selection, the forest rule neighborhood, and variable weighting. Second, benchmark experiments compare the method with existing OOD detectors across synthetic, regression, and high-dimensional survival settings. Third, an esophageal cancer case study shows how the scores inform a clinical survival analysis. Software implementing the method is available in the CRAN package `varPro`. Public benchmark datasets and microarray data used in the experiments are available from the cited references.

1.4 Organization

The paper is organized as follows. Section 2 reviews related work and explains how our approach differs. Section 3 develops the methodology, and Section 4 describes the comparison methods. Section 5 presents the benchmarking results. Section 6 applies the method to a survival analysis of esophageal cancer. Section 7 summarizes and discusses limitations.

2 Related work

Research on out-of-distribution (OOD) behavior spans anomaly detection, open-set recognition, selective prediction, domain generalization, and OOD detection [2, 11–13]. Recent surveys distinguish methods developed for supervised learning [2, 11, 12] from methods tailored to structured domains such as graphs and time series [14–16]. Our focus is test time OOD scoring for supervised tabular models, especially regression and survival analysis, where the score should use what the fitted model has learned about prediction relevant covariates.

2.1 Dataset shift and prediction relevance

Dataset shift is commonly described as a difference between the joint training and test distributions. Standard taxonomies distinguish changes in the covariate distribution from changes in the conditional law of the outcome [3, 4, 7]. Covariate shift refers to a change in $\mathbb{P}_X$ with $\mathbb{P}_{Y|X}$ stable, while concept shift refers to a change in the conditional relationship between predictors and outcome [5–7]. Domain adaptation theory gives related bounds for learning when training and test distributions differ [8].

In supervised tabular prediction, a covariate departure is most relevant when it occurs in directions that affect the fitted prediction rule. Departures confined to nuisance coordinates may make a point unusual under $\mathbb{P}_X$ without making it unusual for the prediction task. This motivates an OOD score that uses the fitted model to determine both the variables and the local training cases used in the comparison.

2.2 Model based OOD scores

Most OOD detection methods were developed for classification, where class probabilities, logits, and learned representations provide natural scoring rules. Examples include maximum softmax probability [1], ODIN [17], energy-based scores [18], Mahalanobis scores in learned feature space [19], deep nearest neighbors [20], and posterior or representation shaping methods [21–23]. Related subspace and spectral methods decompose or project learned

activations before scoring [24–26]. These approaches can be highly effective for image and text benchmarks, but they are closely tied to discrete labels, deep representations, or global feature spaces, and are not directly designed for continuous outcomes or survival data.

For regression and related continuous outcome problems, OOD detection has often followed an uncertainty based paradigm. Deep ensembles [27], Monte Carlo dropout [28], evidential regression [29], prior networks [30], predictor entropy methods [31], Gaussian processes, and deep kernel learning [32–34] use predictive uncertainty as an OOD signal. These methods are model-aware, but the resulting scores are usually global in the feature space and do not explicitly separate departures in predictive variables from departures in nuisance variables.

2.3 Tabular, geometric, and calibrated detectors

A separate line of work screens for unusual test inputs using the covariate distribution alone. Standard tabular baselines include distance based outlier scores and k-nearest-neighbor criteria [35, 36], local outlier factor [37], one-class support vector machines [38], and tree partition scores such as Isolation Forest [39]. These methods are useful general anomaly detectors, but they do not use the supervised response and therefore do not distinguish departures in variables that matter for prediction from departures in variables that do not.

Recent tabular and clinical studies highlight the practical importance of this issue. ODROP trains a separate OOD reject option for disease onset prediction [40], while the Chameleons benchmark studies OOD behavior in medical tabular data [41]. Broader medical OOD reviews also emphasize the need for accurate and interpretable detection in clinical settings [42, 43]. These studies show the importance of OOD detection for tabular risk prediction, but they do not provide a model-aware, subspace-aware score for regression or survival models.

Conformal methods offer a complementary perspective. Given a nonconformity score, conformal prediction can provide calibrated p-values, intervals, or testing procedures [44–48]. Such methods can calibrate OOD decisions once a score has been chosen. Our focus is different. We design the score itself, and leave conformal calibration of the proposed score for future work.

2.4 Positioning

Most existing methods emphasize one of two properties. Some are model-aware because the score depends on a fitted predictor trained on labeled data. Others are subspace-aware because the score is computed after restricting attention to selected variables or learned feature directions. In the tabular regression and survival settings considered here, these properties are rarely combined. Classification and deep representation methods are often model-aware but are not designed for continuous outcomes or survival analysis. Classical tabular anomaly detectors work directly with X and are not model-aware. Uncertainty based regression methods use the fitted model, but usually score globally in the full feature space.

The proposed method is designed for this gap. It is model-aware because the score is built from a supervised forest trained on labeled data, and subspace-aware because the final comparison is restricted to variables selected as task relevant. The next section gives the construction.

3 Methodology

We now give the construction of the proposed method. Given labeled training data $(x_1, y_1), \ldots, (x_n, y_n)$, we first fit a random forest [49]. For a test input x, the forest is then used to construct two objects needed for scoring. The first is a signal set $S \subset \{1, \ldots, d\}$ defining the coordinates on which x is compared with the training data. The second is a local reference neighborhood $\mathcal{N}_K(x)$ defining the training cases to which x is compared. The final score averages a distance from x to this neighborhood using only the coordinates in S.

Random forests are a natural choice for our approach because they provide a common framework for regression, classification, and survival outcomes, and because their learned partitions group observations with similar outcome behavior. This partition based notion of similarity is commonly summarized by random forest proximity, where two inputs are considered similar if they often fall in the same terminal nodes across trees. Our construction however uses the forest differently. Rather than using terminal node co-occurrence itself as the OOD score, we use the learned rule structure of the forest to build the signal set and the local reference neighborhood.

Each tree can be written as a collection of decision rules. A rule is a path from the root to a terminal node, expressed as simple conditions on individual variables, for example $x^{(1)} > 3$ and $x^{(5)} < 1$. These conditions define a rectangular region of the input space. Across the ensemble, the full collection of rules describes how regions of the covariate space are associated with outcome patterns. The rules are used in two connected steps. First, they are analyzed to identify variables carrying predictive information, yielding a signal set $S \subset \{1, \ldots, d\}$. Second, for a test input x, the rules that contain x are relaxed one selected variable at a time to identify training points that repeatedly co-occur with x under the forest. These co-occurrences rank the training points and form the reference neighborhood $\mathcal{N}_K(x)$. The final OOD score is the average distance from x to this neighborhood, restricted to the signal coordinates, $d(x) = \sum_{z \in \mathcal{N}_K(x)} D_S(x, z)/K$.

The remainder of this section develops each component in turn. A standalone summary is given in Algorithm 1, which we call *OutPro* (OOD using variable priority). For a test input $x = (x^{(1)}, \ldots, x^{(d)}) \in \mathbb{R}^d$, OutPro takes the labeled training data and trained forest as input and returns a scalar OOD score $d(x)$, with larger values indicating greater departure from the training distribution in the predictive subspace.

3.1 Variable priority and selection of *S*

Both neighborhood construction and distance evaluation depend on a set of task relevant variables

$$S = \{s_1, \ldots, s_q\} \subset \{1, \ldots, d\}.$$

We estimate S, together with weights $\{w_s\}_{s \in S}$ measuring the relative strength of each selected coordinate, using Variable Priority (VarPro) [50–53]. VarPro is a model based feature selection method derived from the rule regions produced by a forest. For a rule region, the release region for variable s is the expanded region obtained by removing the constraint on s while keeping all other bounds fixed.

A variable receives high priority when releasing its constraint consistently changes the outcome behavior of the training points in the region. Variables whose release has little effect are treated as noise. Thus VarPro scores variables by outcome changes under rule relaxation, rather than by split frequency or tree usage alone. This helps distinguish genuine predictors from correlated surrogates and is directly relevant for OOD scoring. Including noise variables

in S dilutes the distance calculation across uninformative dimensions, whereas excluding signal variables can hide departures in the predictive subspace.

3.2 Frequency profiling via rule relaxation

The same release operation is used to build the reference neighborhood for a test input x. Let $\mathcal{R}(x)$ denote the set of decision rules from the forest whose terminal regions contain x. Each rule $\zeta \in \mathcal{R}(x)$ defines a hyperrectangle $R(\zeta) \subset \mathbb{R}^d$. For each selected coordinate $s \in S$, let $R(\zeta^s)$ be the release region formed by removing the constraint on s and retaining all other bounds.

Algorithm 1: OOD using variable priority (OutPro)

Input: Training data $\{(x_i, y_i)\}_{i=1}^n$; test point x; number of neighbors K; distance function D.
Output: OOD distance score $d(x)$.

Step 1: Train a supervised random forest
Fit a random forest model to $\{(x_i, y_i)\}$ and extract decision rules $\{\zeta_j\}$ for terminal nodes.
Step 2: Apply VarPro to identify signal variables
Obtain the signal set $S = \{s_1, \ldots, s_q\} \subset \{1, \ldots, d\}$ and the importance values $\{w_s\}_{s \in S}$ using VarPro.
Step 3: Identify rules containing the test point
Let $\mathcal{R}(x) = \{\zeta_j : x \in R(\zeta_j)\}$ be the set of rules whose regions contain x.
Step 4: Compute relative frequencies
for each *training point* x_i **do**
  for each $s \in S$ **do**
    Set $n_s(x_i) = 0$.
    for each $\zeta \in \mathcal{R}(x)$ **do**
      Construct the release region $R(\zeta^s)$ by removing the constraint on variable s.
      if $x_i \in R(\zeta^s)$ **then**
        Update $n_s(x_i) \leftarrow n_s(x_i) + 1$.
  Compute $n(x_i) = \sum_{s \in S} n_s(x_i)$.
  if $n(x_i) = 0$ **then**
    set $p_s(x_i) = 0$ for all $s \in S$ and continue.
  Compute $p_s(x_i) = n_s(x_i)/n(x_i)$ for all $s \in S$.
  Compute $G(x_i) = \sum_{s \in S} p_s(x_i)\big(1 - p_s(x_i)\big)$.
  Compute $W(x_i) = n(x_i) \cdot G(x_i)$.
Step 5: Select the K highest-scoring neighbors
Sort $\{x_i\}$ in decreasing order of $W(x_i)$ and select the top K neighbors, $\mathcal{N}_K(x)$.
Step 6: Compute the OOD distance
for each $z \in \mathcal{N}_K(x)$ **do**
  Standardize the coordinates using training means and variances.
  Compute $D(x, z)$ using the selected distance function and (if appropriate) the normalized importance weights $\{w_s\}_{s \in S}$.
Return $d(x) = \frac{1}{K} \sum_{z \in \mathcal{N}_K(x)} D(x, z)$ for pairwise metrics, or $d(x) = d_{\text{optics}}(x)$ when OPTICS reachability is used.

Fig. 1 Release-region mechanism used for frequency profiling. The green cube is a rule containing the test point x shown in black, and grey cuboids are release regions formed by removing one coordinate constraint at a time. Training points A and B are equidistant from x in ordinary Euclidean distance, but their release profiles differ. A appears in all three release regions, giving counts (1, 1, 1), whereas B appears only after releasing $x^{(2)}$ and $x^{(3)}$, giving counts (0, 1, 1). Thus A is more similar to x under the rule relaxation profile

For each training point x_i, we count its appearances in these release regions. For $s \in S$, define

$$n_s(x_i) = \sum_{\zeta \in \mathcal{R}(x)} I\{x_i \in R(\zeta^s)\},$$

and let

$$n(x_i) = \sum_{s \in S} n_s(x_i).$$

When $n(x_i) > 0$, the frequency profile of x_i relative to x is

$$(p_{s_1}(x_i), \ldots, p_{s_q}(x_i)) = \left(\frac{n_{s_1}(x_i)}{n(x_i)}, \ldots, \frac{n_{s_q}(x_i)}{n(x_i)} \right).$$

If $n(x_i) = 0$, we set $p_s(x_i) = 0$ for all $s \in S$ and exclude x_i from neighborhood construction. The profile records how often a training point co-occurs with x after releasing each selected coordinate, and therefore summarizes its alignment with x under the forest's local rule structure.

Figure 1 illustrates this construction for a single rule and three selected coordinates. Points A and B are equidistant from x in ordinary Euclidean distance, but their release profiles differ. Point A appears in all three release regions, whereas point B appears in only two. The frequency profile therefore captures local similarity under the forest's rule structure, rather than under a fixed Euclidean metric.

3.3 Proximity scoring from frequency dispersion

This subsection defines a proximity score $W(x_i)$ for each training point x_i, relative to the test input x. The score is used to rank training points before the final OOD distance is computed. A training point receives a large value of $W(x_i)$ when it appears often in the release regions for x and when those appearances are spread across the selected variables in S. A point receives a smaller value when it appears rarely, or when its appearances are concentrated in release regions for only a few selected variables.

For a training point x_i with $n(x_i) > 0$, recall that $p_s(x_i)$ is the relative frequency with which x_i appears after releasing coordinate $s \in S$. We summarize the balance of this profile using the Gini impurity

$$G(x_i) = \sum_{s \in S} p_s(x_i)\big(1 - p_s(x_i)\big).$$

Larger values occur when co-occurrence is spread more evenly across the selected coordinates. The final proximity score also accounts for the total number of appearances,

$$W(x_i) = n(x_i)G(x_i).$$

Thus $W(x_i)$ favors training points that both appear frequently in release regions and have balanced co-occurrence across the predictive coordinates.

3.4 OOD distance via subspace distance functions

The reference neighborhood for x is the set $\mathcal{N}_K(x)$ of the K training points with the largest proximity scores $W(x_i)$. The OOD score is then the average distance from x to this neigh-

Table 1 Distance functions used in the experiments. Coordinates are standardized using training means and variances, and all distances are restricted to the selected coordinates S. Except for Mahalanobis distance, each coordinate $s \in S$ is weighted by its normalized VarPro weight w_s. The first five rows define pairwise distances $D(x, z)$ used in the averaged OOD score, while OPTICS defines a direct reachability score for x

Distance	Definition
Product	$D_{\text{prod}}(x,z) = \prod_{s\in S}\left(\lvert x^{(s)} - z^{(s)}\rvert + \varepsilon\right)^{w_s}$, with $\varepsilon > 0$ used for numerical stability.
Euclidean	$D_{\text{euclid}}(x,z) = \left\{\sum_{s\in S} w_s (x^{(s)} - z^{(s)})^2\right\}^{1/2}$.
Manhattan	$D_{\text{manh}}(x,z) = \sum_{s\in S} w_s \lvert x^{(s)} - z^{(s)}\rvert$.
Minkowski	$D_{\text{mink}}(x,z) = \left\{\sum_{s\in S} w_s \lvert x^{(s)} - z^{(s)}\rvert^p\right\}^{1/p}$. We use $p = 4$ in the empirical studies.
Mahalanobis	Let $\Sigma^{(S)}$ be the empirical covariance matrix of the training data restricted to S. Then $D_{\text{mahal}}(x,z) = \left\{(x^{(S)} - z^{(S)})^T (\Sigma^{(S)})^{-1}(x^{(S)} - z^{(S)})\right\}^{1/2}$.
OPTICS	OPTICS [54] is applied to $\mathcal{N}_K(x) \cup \{x\}$ after restricting to S and scaling coordinate s by $\sqrt{w_s}$. The OOD score is the reachability value $d_{\text{optics}}(x) = \text{Reachability}\big(x \mid \mathcal{N}_K(x), \texttt{minPts}, \{w_s\}_{s\in S}\big)$. We use the `optics` function from the R package `dbscan` with `minPts`=5 [55].

borhood,

$$d(x) = \frac{1}{K}\sum_{z\in\mathcal{N}_K(x)} D(x,z).$$

The distance $D(x, z)$ is computed only on the selected variables $s \in S$ and is weighted by the VarPro scores $\{w_s\}_{s\in S}$. The weights are positive and normalized, so higher priority variables receive greater emphasis and lower priority variables are down-weighted. Several different types of distance functions are used in the empirical experiments; these are conveniently summarized in Table 1.

4 Comparison methods

We compare OutPro with OOD detectors that are applicable to continuous outcomes and have reliable open source implementations. The baselines include uncertainty based methods, latent representation methods, gradient based scores, and classical unsupervised tabular detectors. The unsupervised methods use only the training covariates, whereas the model based methods use a fitted predictor. Table 2 summarizes the score and implementation used by each comparison method.

All neural network methods share the same base architecture, implemented with the Keras API [56]. The network has two hidden ReLU layers, batch normalization, dropout, and a single output node trained by mean squared error. MSP, ODIN, Energy, Sensitivity, DKL, GMM, and latent space k-nearest neighbors are computed from this shared fitted network, as described in Table 2. Conformal prediction and deep ensembles use the same architecture

Table 2 Comparison methods. Here $f(x)$ is the fitted prediction, $h(x)$ is the penultimate layer representation of the neural network, $\bar{y}_{\text{train}}$ is the mean training response, and $\tilde{x} = x + \epsilon\,\text{sign}(x)$ with $\epsilon = 0.01$. Input space methods use standardized covariates unless noted otherwise.

Method	OOD score or implementation summary
MSP [1]	Regression adaptation using prediction deviation, $\lvert f(x) - \bar{y}_{\text{train}}\rvert$. This is monotone equivalent to the reciprocal confidence score used in the implementation.
ODIN [17]	Regression adaptation using the perturbed input $\tilde{x}$ and score $\lvert f(\tilde{x}) - \bar{y}_{\text{train}}\rvert$.
Energy [18]	Perturbation based latent score $\lVert h(\tilde{x})\rVert_2^2$.
Mahalanobis	Input space Mahalanobis distance $\{(x-\mu)^T \Sigma^{-1}(x-\mu)\}^{1/2}$, where μ and Σ are estimated from the training covariates.
LOF [37]	Local outlier factor computed on standardized covariates with $k = \min(20, n-1)$.
Isolation Forest [39]	Isolation Forest anomaly score using 200 trees and subsample size $\min(256, n)$.
OCSVM [38]	One class support vector machine with radial kernel $K(x,x') = \exp\{-\gamma\lVert x - x'\rVert_2^2\}$, $\gamma = 1/d$. The OOD score is the negative fitted decision function. The parameter ν is set to the benchmark cutoff rate, truncated to $[1/\max(2,n), 0.5]$.
kNN density [35, 36]	Robust k-nearest-neighbor density score. Covariates are centered by the training median and scaled by the training median absolute deviation. The score is the log median radius to the $k = \min(20, n-1)$ nearest training points.
Conformal prediction [44]	Split conformal score using 75% of the training data for fitting and 25% for calibration. The ranking score is $\lvert f(x) - \text{median}(f_{\text{calib}})\rvert$; the binary decision compares this value with the corresponding calibration quantile.
Gaussian process	Local approximate Gaussian process regression using `laGP`; the OOD score is the posterior predictive variance at x.
Sensitivity [58]	Average absolute input gradient, $d^{-1}\sum_{j=1}^{d} \lvert\partial f(x)/\partial x^{(j)}\rvert$.
DKL [33]	Deep kernel learning baseline. A Gaussian process is fit by `laGP` to the neural representation $h(x)$, and the OOD score is the posterior predictive variance.
Deep ensembles [27]	Variance of predictions across independently initialized neural network ensemble members.
GMM [59]	Gaussian mixture density in PCA reduced latent space. The OOD score is $-\log p(\tilde{h}(x))$, where $\tilde{h}(x)$ is the reduced representation.
DkNN [20]	Latent space k-nearest-neighbor score, $k^{-1}\sum_{j=1}^{k} \lVert h(x) - h(x_{(j)})\rVert_2$, using nearest neighbors in the neural representation.

but refit it in their own split calibration and ensemble settings. Gaussian process methods use the `laGP` package [32, 57]. All scores are oriented so that larger values indicate stronger evidence of OOD behavior.

5 Benchmarking OOD detection methods in regression

We evaluate performance using a combination of simulated and real regression problems. On synthetic data, we use a well-known machine learning model with independent covariates (Sect. 5.1), which provides a controlled anomaly framework with clearly defined ground-truth

labels. For real data, we consider two sets of experiments. The first is a large collection of regression benchmarks (Sect. 5.3) with varying sample sizes, dimensions, and signal-to-noise ratios. The second uses high-dimensional microarray survival studies (Sect. 5.4), where the number of features greatly exceeds the sample size. For the real-data experiments, anomalies are generated using a copula-based strategy described in Sect. 5.2, which allows targeted perturbations of marginal distributions and joint dependence. Test cases are drawn from a held-out subset of the data, and anomalous inputs are created by perturbing selected features to simulate OOD behavior.

5.1 Friedman simulation

We first consider a simulated regression setting based on the well-known Friedman model [60]. Each covariate $X^{(j)} \sim U[0, 1]$ independently for $j = 1, \ldots, d$. The response is generated according to

$$Y = 10 \sin\left(\pi X^{(1)} X^{(2)}\right) + 20\left(X^{(3)} - 0.5\right)^2 + 10X^{(4)} + 5X^{(5)} + \varepsilon,$$

where $\varepsilon \sim N(0, \sigma^2)$ and the remaining $d - 5$ covariates are irrelevant noise features. We use $d = 20$ and $\sigma = 1$.

To simulate OOD instances, we apply a fixed additive shift to a subset of features identified as predictive of the outcome. Although the true signal variables are known in this simulation, we use tree-based gradient boosting [61] to rank features by their contribution to predictive accuracy. This data driven approach to feature selection mirrors the methodology used in our later real-data experiments and ensures consistency across settings. From the ranked list, the top ten percent of features are selected, and each is perturbed by a fixed additive amount scaled by its standard deviation, with direction randomly assigned for each test point.

5.1.1 Ground truth

The additive shift procedure is a common strategy in OOD detection research. However, the labeling rule that defines a shifted point as anomalous has a subtle issue that is often overlooked: a shifted point may lie entirely within the support of the training distribution and therefore represent a perfectly valid input.

To see this, note that under the Friedman setup the covariates $X^{(j)}$ are independent and uniformly distributed on $[0, 1]$. Applying an additive shift $\delta = (\delta_1, \ldots, \delta_d)$ to the covariate vector $X = (X^{(1)}, \ldots, X^{(d)})$ produces a perturbed point

$$X^* = X + \delta.$$

Because the covariates are independent with bounded support, a shifted point X^* becomes anomalous only if at least one coordinate falls outside the unit interval. For coordinate j, the condition $X^{(j)} + \delta_j \in [0, 1]$ is equivalent to

$$-\delta_j \le X^{(j)} \le 1 - \delta_j.$$

Hence the probability that all coordinates remain within the original hypercube is

$$\mathbb{P}(\text{all coordinates inside}) = \prod_{j=1}^{d} (1 - |\delta_j|)_+,$$

where $(a)_+ := \max(a, 0)$. The true fraction of shifted points that fall outside the support is therefore 1 minus this value, which for small shifts can be approximated to first order as $\sum_{j=1}^{d} |\delta_j|$ and can therefore be substantially less than 1.

Therefore many nominally shifted points may still lie entirely within the original support and are therefore legitimate data values, particularly when shifts are small or the dimension is moderate. To avoid this ambiguity, we adopt the rule that a point is labeled anomalous if and only if at least one of its coordinates falls outside [0, 1].

5.1.2 Experimental settings

We generated 100 independent datasets of size $n = 2000$ from the Friedman model. Each dataset was randomly split into 80% training and 20% testing and anomalies were created by perturbing test points using shifts as described above. Shifts were applied to the top predictive features ranked by gradient boosting and scaled by the empirical standard deviation of each feature. We considered five shift magnitudes: 0.05, 0.25, 0.5, 1.0, and 2.0. Ground-truth labels were defined by classifying a perturbed test point as anomalous if and only if at least one coordinate fell outside [0, 1].

5.1.3 Results

Performance was measured by the area under the precision-recall curve (AUC-PR). Boxplots in Figure 2 show that the OutPro procedures are among the best for small shifts (0.05 to 0.5), which is the setting where anomalies are subtle and remain close to the training distribution. To systematically compare methods, critical difference (CD) plots are given in Figure 3, with significance assessed using the Nemenyi test. The OutPro product score attains the best average rank at every shift magnitude, and Manhattan is consistently second. For the smallest shifts, the best baselines are the classical tabular methods, especially Isolation Forest, OCSVM, robust kNN density, LOF, and the global Mahalanobis score, but they do not overtake OutPro. As the shift grows, GP and DKL (GP) improve and, together with DkNN, become competitive with OutPro. Isolation Forest is less effective for the larger shifts, while robust kNN density and LOF remain in the middle. Overall, the results show that density based tabular baselines can be competitive when the shift is large enough to create obvious low density points, but OutPro is best when the perturbation is small and functionally targeted.

5.2 Copula-based strategy for anomaly generation

The Friedman simulation provides a controlled benchmark with independent features and bounded support. Real tabular data are more complex, with skewed or heavy-tailed marginals and nontrivial dependence among variables. To generate realistic anomalies in this setting, we use a copula-based strategy that separates marginal behavior from dependence and perturbs each component in a controlled way.

Let $X = (X^{(1)}, \ldots, X^{(d)})$ be a continuous random vector with marginal CDFs $F_1, \ldots, F_d$. Applying the probability integral transform coordinatewise gives

$$U^{(j)} = F_j(X^{(j)}), \quad j = 1, \ldots, d.$$

For continuous marginals, each $U^{(j)}$ is uniform on [0, 1], and the joint distribution of $U = (U^{(1)}, \ldots, U^{(d)})$ is the copula C. By Sklar's theorem [62],

$$F(x^{(1)}, \ldots, x^{(d)}) = C\{F_1(x^{(1)}), \ldots, F_d(x^{(d)})\}.$$

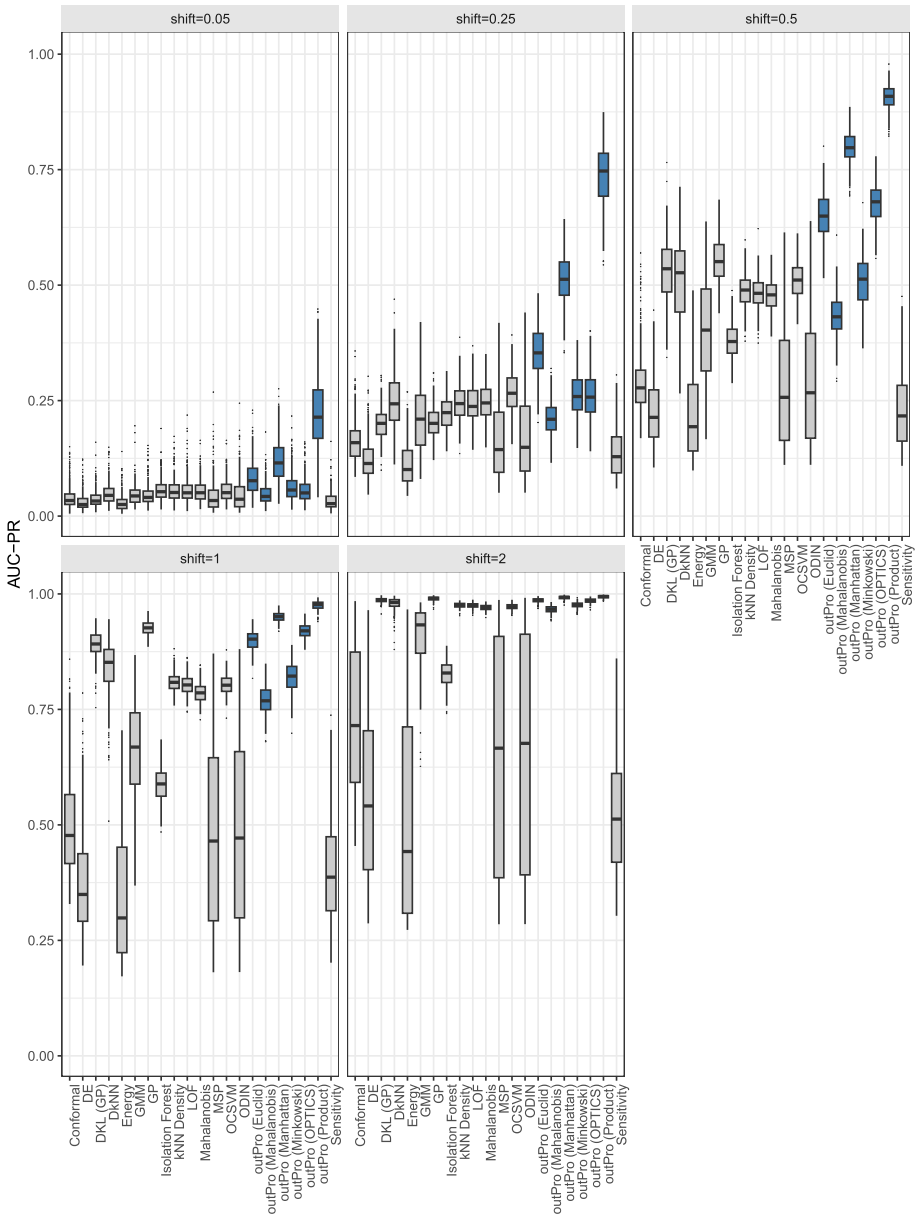

Fig. 2 Boxplots of AUC-PR values for each method across 100 runs at varying shift magnitudes in the Friedman simulation. Blue indicates OutPro methods. Performance improves for all methods as the shift magnitude increases. The OutPro product and Manhattan scores are best for small shifts (0.05 to 0.5), where detection is most difficult

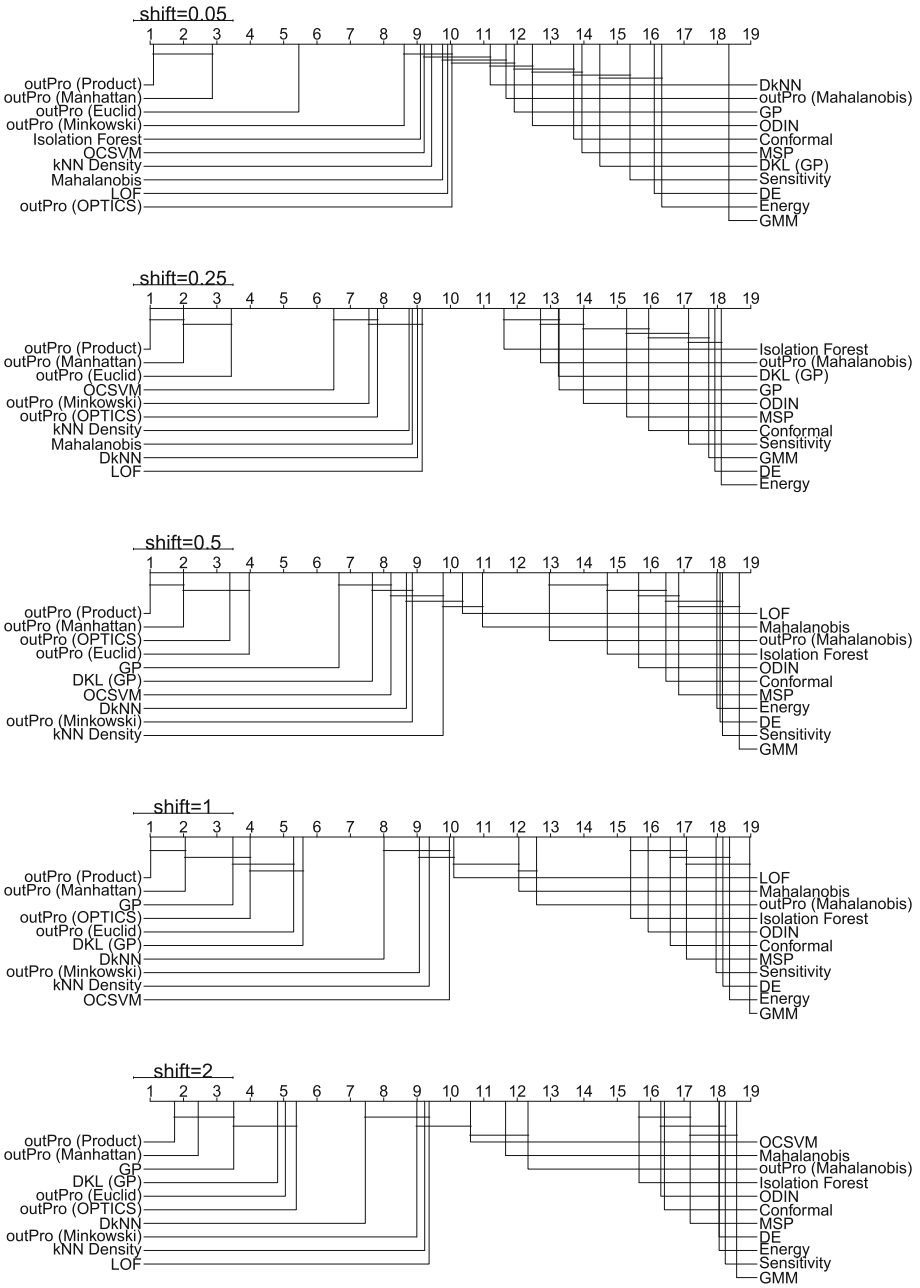

Fig. 3 Critical difference (CD) plots comparing average AUC-PR rankings for the Friedman simulation at each shift magnitude. Lower ranks indicate better performance. Methods with no statistically significant difference at $\alpha = 0.05$ are connected by a horizontal bar. For shifts 0.05, 0.25, and 0.5, the OutPro product and Manhattan scores rank near the top, while the best classical tabular baselines are in the next tier. The product score has the best average rank at every shift magnitude

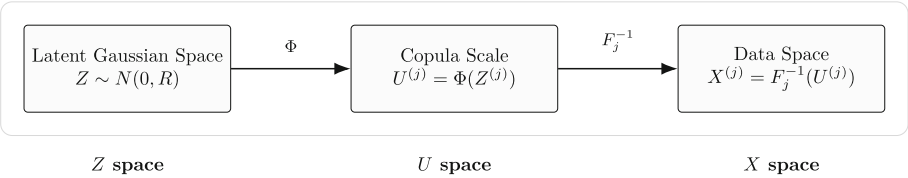

Fig. 4 Three-stage transformation used by the copula anomaly generation procedure. A latent Gaussian vector Z determines dependence, the transformation $U = \Phi(Z)$ places each coordinate on a common uniform scale, and the inverse marginal CDFs map back to the data space X. Perturbing different stages of this pipeline yields the `warp`, `joint`, and `support` anomaly modes

Thus a point on the copula scale can be mapped back to the data scale by $x^{(j)} = F_j^{-1}(u^{(j)})$. The marginal inverse CDFs determine the coordinate scales, while the joint arrangement of the u values determines dependence.

We model dependence with a Gaussian copula [63, 64]. A latent Gaussian vector $Z = (Z^{(1)}, \ldots, Z^{(d)})$ is generated as $Z \sim N(0, R)$, where R is estimated from the training data, and $U^{(j)} = \Phi(Z^{(j)})$, $j = 1, \ldots, d$. This gives the three-stage transformation (Figure 4)

$$Z \longrightarrow U \longrightarrow X,$$

where Z controls dependence, U places coordinates on a common uniform scale, and the inverse marginals map back to the observed data space. Perturbing different stages of this pipeline yields the three anomaly modes summarized in Table 3.

The three anomaly modes are designed to test different ways a tabular distribution can depart from the training data. The `warp` mode changes marginal tail behavior by distorting the uniform variables before mapping back to the original coordinate scales. This produces observations that remain tied to the estimated dependence structure but have altered marginal behavior. The `joint` mode perturbs the latent Gaussian representation, creating points in atypical regions of the dependence structure while preserving the original marginal mapping. The `support` mode moves a latent training point outward and then maps it through modified inverse CDFs that extrapolate beyond the observed coordinate ranges.

5.3 Benchmarking on diverse real and simulated datasets

For benchmarking we used a curated subset of regression datasets from the Penn Machine Learning Benchmark (PMLB) repository [65, 66]. Only datasets with at least 10 features and a continuous response were retained, resulting in 61 datasets with dimensions ranging from $d = 10$ to 124 and sample sizes from $n = 47$ to 1066.

5.3.1 Experimental settings

Each dataset was randomly split into 80% training and 20% testing. Synthetic anomalies were generated using the copula-based procedure described in Section 5.2, with the number of anomalies matched to the number of test points. Each copula mode (`warp`, `joint`, `support`) was applied independently. To ensure that perturbations produced meaningful distributional shifts, the copula transformations were applied only to features identified as predictive of the outcome by selecting the top 10% using gradient boosting, as in the Friedman simulation, while the remaining features were left unchanged. Performance was measured by AUC-PR, with all evaluations repeated over 100 independent runs.

Table 3 Copula anomaly modes used in the experiments

Mode	Generation and labeling
`warp`	Draw $z \sim N(0, R)$ and set $u^{(j)} = \Phi(z^{(j)})$. Apply the coordinate-wise warp $u^{*(j)} = \{u^{(j)}\}^\gamma / [\{u^{(j)}\}^\gamma + \{1-u^{(j)}\}^\gamma]$ with $\gamma > 1$, and map back by $x^{*(j)} = F_j^{-1}(u^{*(j)})$. The label is $a^* = 1\{d_M(x^*) > \tau_{1-q}\}$, where $z^* = \Phi^{-1}(u^*)$ and $d_M(x^*) = \{z^{*\top} R^{-1} z^*\}^{1/2}$.
`joint`	Choose a latent training point z_b, draw a random unit vector ζ, and sample $r = \sqrt{Q}$ with $Q \sim \chi^2_d$ truncated to $[\chi^2_d(1-q), \infty)$. Set $z^* = z_b + r\zeta$ and map back by $x^{*(j)} = F_j^{-1}\{\Phi(z^{*(j)})\}$. These points perturb the dependence structure while preserving the marginal mapping, and are labeled anomalous, $a^* = 1$.
`support`	Choose a latent training point z_b with $r_b = \{z_b^T R^{-1} z_b\}^{1/2}$. Draw $r^* = \sqrt{Q^*}$ from the upper $(1-q)$ tail of χ^2_d and set $z^* = z_b(r^*/r_b)$. Map back using a modified inverse CDF, $x^{*(j)} = \tilde{F}_j^{-1}\{\Phi(z^{*(j)})\}$, which extrapolates beyond the empirical support. These points are labeled anomalous, $a^* = 1$.

Here $z_i = \Phi^{-1}\{F(x_i)\}$ denotes the latent Gaussian representation of training point x_i, q is the tail probability, and τ_{1-q} is the empirical $(1-q)$ percentile of $d_M(x_i) = \{z_i^T R^{-1} z_i\}^{1/2}$ over the training data (all experiments use $q = 0.05$). The value a^* is a binary label with value 1 for anomalous data

5.3.2 Results

Given the large number of datasets, we summarize performance by the average AUC-PR rank across datasets, shown as critical difference (CD) plots in Figure 5. Under the `warp` mode, the best methods are OCSVM, Isolation Forest, LOF, the global Mahalanobis score, robust kNN density, and DkNN. Next are the OutPro procedures, with Mahalanobis performing the best, followed by Euclidean, Minkowski, and Manhattan, while the product and OPTICS versions are lower. This pattern is reasonable because `warp` creates marginal tail distortion within the observed support and therefore favors methods driven by density, isolation, or global geometry.

Under the `joint` mode, the pattern changes significantly. The OutPro procedures are the best, with Manhattan first, followed by Euclidean, product, Minkowski, OPTICS, and Mahalanobis. The best non-OutPro methods are OCSVM and LOF, with energy, robust kNN density, and the global Mahalanobis score forming the next group. This shows that when anomalies alter dependence, methods built from local predictive profiles perform better than generic full-space scoring rules.

The same pattern holds for the `support` mode. OutPro procedures are again the best. Among the non-OutPro methods, global Mahalanobis score is the best, followed by OCSVM, LOF, robust kNN density, and energy, while conformal and Isolation Forest are in the next group. This shows that when anomalies exist outside the observed support, methods derived from local predictive profiles are especially effective.

5.4 High-dimensional benchmarking

We next evaluate performance on five high-dimensional microarray datasets commonly used in the survival analysis literature, diffuse large B-cell lymphoma (DLBCL) [67], breast

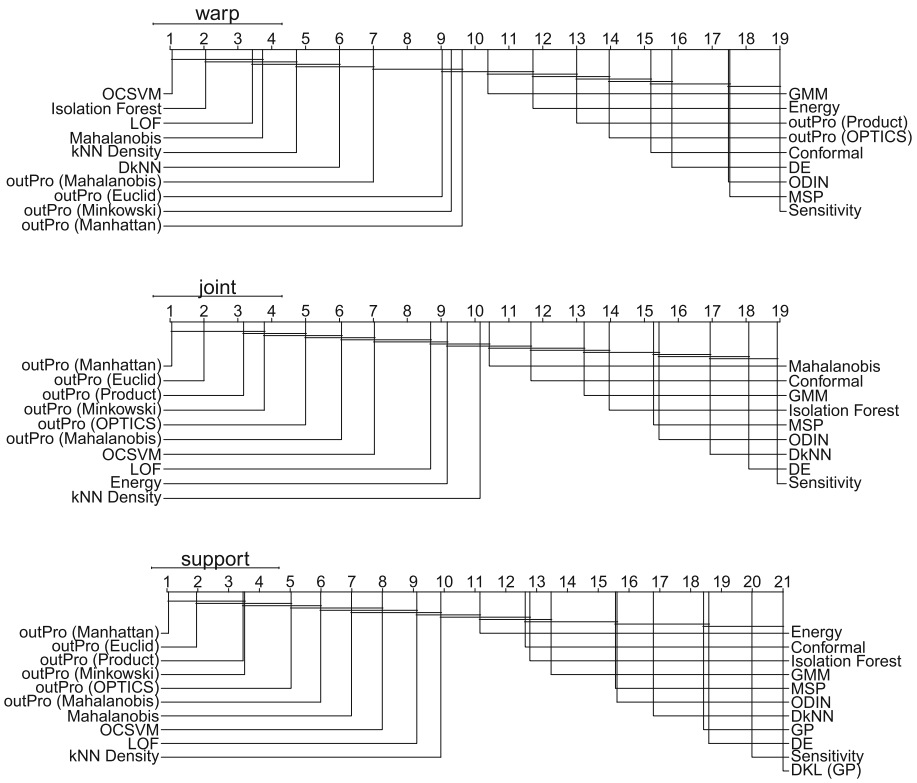

Fig. 5 Critical difference (CD) plots comparing OOD detection methods across 61 PMLB regression datasets. From top to bottom, the panels correspond to the three anomaly generation modes, `warp`, `joint`, and `support`. Methods are ranked by average AUC-PR, and statistically indistinguishable groups at $\alpha = 0.05$ are connected by horizontal bars based on the Nemenyi test

Table 4 Summary of high-dimensional microarray datasets used for benchmarking after Cox-score filtering.

Dataset	n	d
AML	116	629
Breast Cancer	78	475
DLBCL	240	740
Lung Cancer	86	713
MCL	92	881

cancer [68], lung cancer [69], acute myeloid leukemia (AML) [70], and mantle cell lymphoma (MCL) [71].

To place these in a regression framework, we converted survival outcomes into continuous pseudo-responses using random survival forests (RSF) [72]. RSF models were fit to each dataset, and out-of-bag (OOB) mortality predictions were extracted for use as regression targets. Although OutPro can handle survival data directly, this transformation enabled direct comparison with the baseline methods. To improve the signal-to-noise ratio, features were filtered by Cox-score ranking [73]. Table 4 summarizes sample sizes and dimensions for the filtered datasets.

5.4.1 Experimental settings

The experimental design mirrored that of the PMLB analysis. Each dataset was split into 80% training and 20% test sets. Synthetic anomalies were generated using the copula-based procedure under all three modes, with the number of anomalies matched to the number of test points. Anomaly perturbations were restricted to the top 10% of features identified as predictive by gradient boosting, while the remaining features were left unchanged. Each configuration was repeated over 100 independent runs, and performance was measured using AUC-PR. Due to the high-dimensional setting, the global Mahalanobis baseline could not be evaluated directly because the sample covariance matrices were singular, and therefore is omitted.

5.4.2 Results

Boxplots of AUC-PR by dataset and copula mode are shown in Figure 6, with an overall summary via critical difference (CD) plots given in Figure 7. The ranking pattern is more mixed than in the PMLB analysis.

Under the `warp` mode, the best methods are Isolation Forest, robust kNN density, and OCSVM, while LOF and DKL (GP) also perform well. Among the OutPro procedures, Mahalanobis is best, while Manhattan, Euclidean, Minkowski, and product fall somewhere in the middle of all procedures. This again fits the structure of `warp`, which generates tail distorted points that remain largely within the observed support and can therefore be captured well by classical density based rules.

As in the previous PMLB study, the picture changes under the `joint` mode. OutPro Euclidean, Manhattan, Mahalanobis, Minkowski, product, and OPTICS are the top procedures. The best non OutPro methods are robust kNN density, GMM, DkNN, and conformal prediction. Thus when anomalies act through dependence rather than marginal extremes, the local predictive structure used by OutPro is more informative than generic density or uncertainty scores.

Under the `support` mode, the results are mixed. Robust kNN density has the best performance. LOF is also near the top. At the same time, OutPro Mahalanobis, Minkowski, Euclidean, and Manhattan occupy four of the six top spots, showing that OutPro remains highly competitive. The product based OutPro score is less effective, especially under `support`, where it ranks below the other OutPro methods. One explanation is that in these high-dimensional datasets many coordinates carry signal, so a multiplicative combination of coordinate wise discrepancies can amplify noise and dilute informative deviations. Additive or correlation adjusted distances such as Manhattan, Euclidean, Minkowski, and Mahalanobis appear more stable in this setting.

5.5 Sensitivity analysis for K and `minPts`

To assess the stability of OutPro with respect to its main tuning parameters, we carried out a focused ablation on the PMLB regression datasets used in Section 5.3. We chose the `support` copula mode for this analysis because it was the most challenging setting in the benchmark. Recall from Section 3 that K is the number of training points used for the neighborhood $\mathcal{N}_K(x)$. By design, OutPro uses values of K that are much larger than those typical of classical nearest-neighbor methods. The default neighborhood size used in our previous analyses was $K_0 = \min(n/10, 5000)$; here we evaluated K such that K/K_0

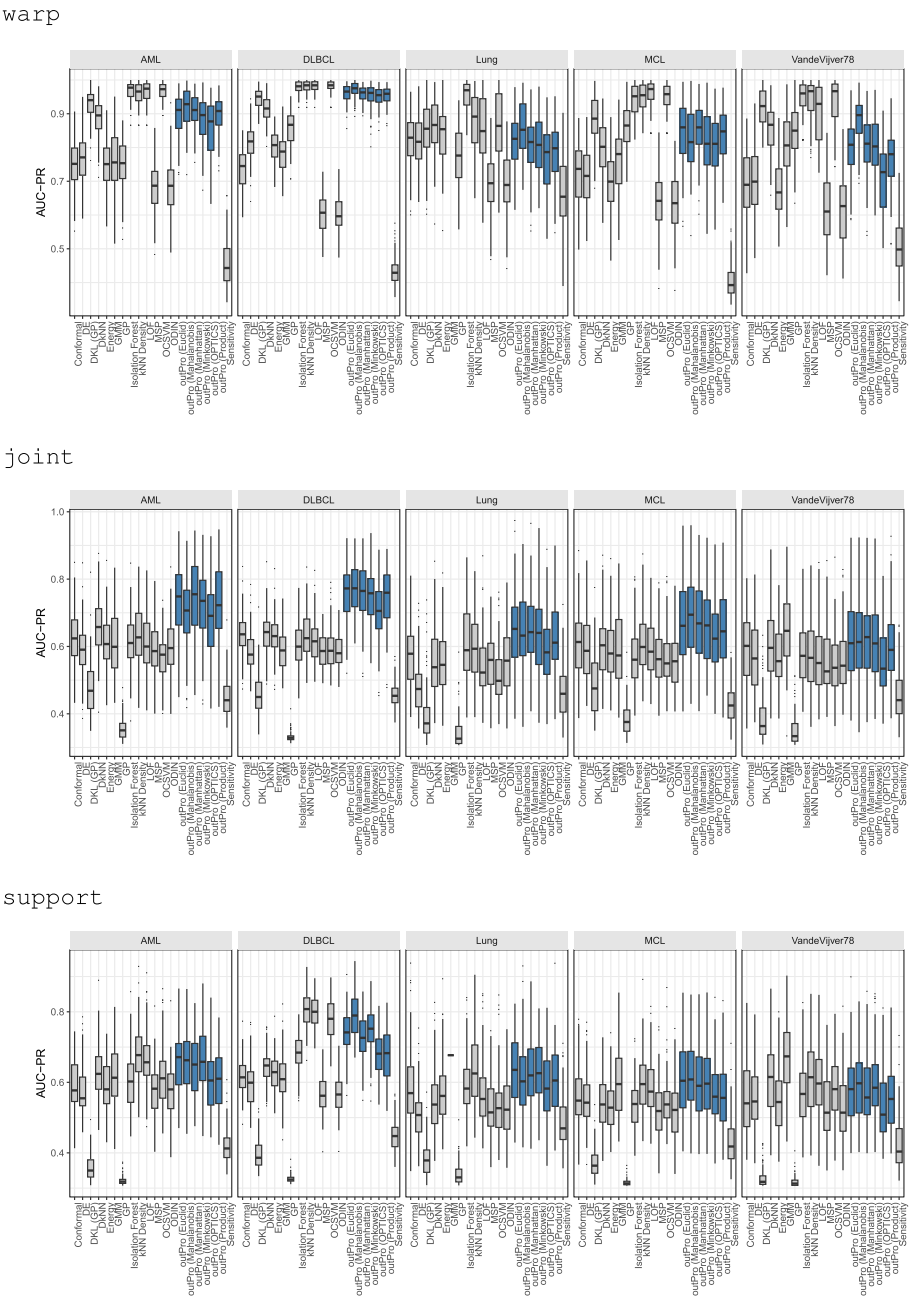

Fig. 6 AUC-PR scores for high-dimensional microarray datasets under the three copula-based anomaly modes, `warp`, `joint`, and `support`. Each box summarizes results over 100 replications. Blue signifies OutPro methods

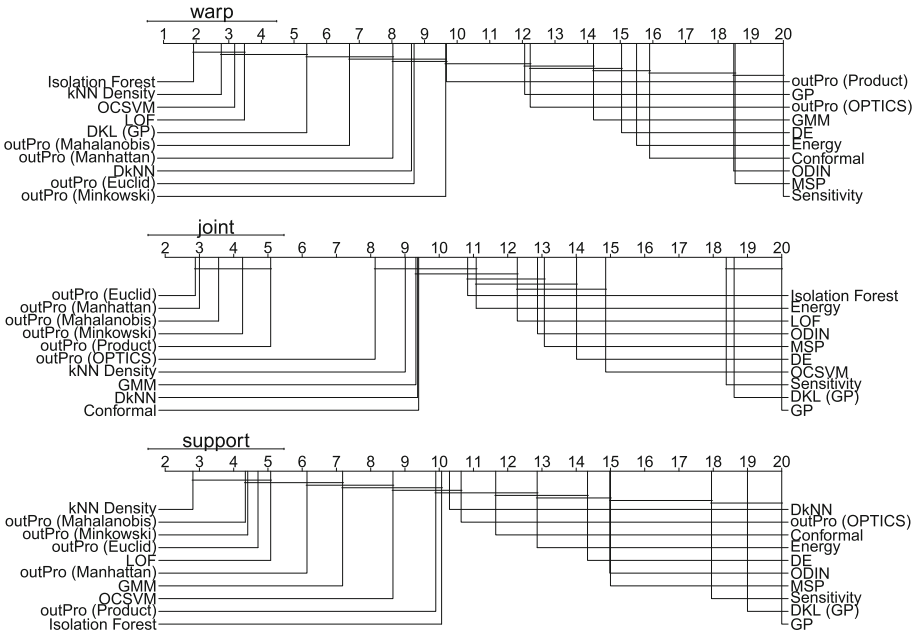

Fig. 7 Critical difference (CD) plots showing average rank performance across the five high-dimensional microarray datasets and copula modes. Lower ranks indicate better performance. The `warp` panel is led by Isolation Forest, robust kNN density, OCSVM, and LOF. The `joint` panel is led by OutPro distances. The `support` panel is more mixed, with robust kNN density and several OutPro distances near the top

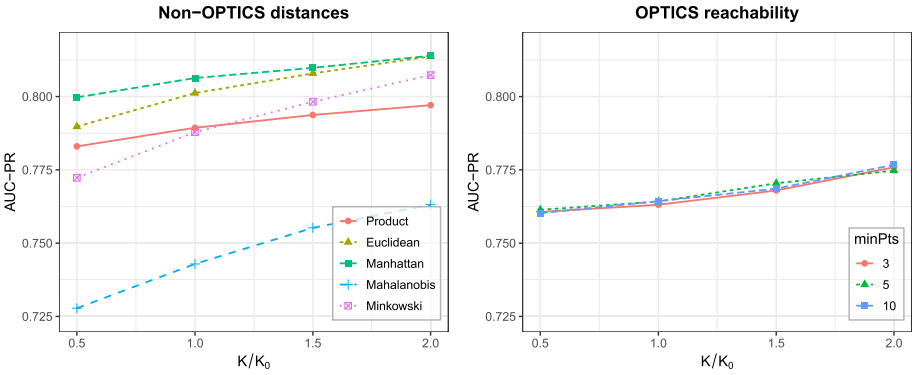

Fig. 8 Sensitivity of OutPro to neighborhood size K under the `support` anomaly mode on the 61 PMLB regression datasets. The left panel shows AUC-PR for the non-OPTICS distances as a function of K/K_0. The right panel shows AUC-PR for the OPTICS reachability score for different values of `minPts`

$\in \{0.5, 1, 1.5, 2\}$. For the OPTICS distance we also varied `minPts` $\in \{3, 5, 10\}$. Because `minPts` enters only the OPTICS reachability calculation, the non-OPTICS distances were summarized after averaging over `minPts`, while OPTICS was reported separately for each value.

Table 5 Component ablation on the PMLB regression benchmark. Lower average AUC-PR rank is better. The column $\bar{r}$ is the average rank across the three copula modes.

Setting	Space	Neighborhood	Weights	`joint`	`support`	`warp`	$\bar{r}$
Full OutPro	$\widehat{S}$	forest	VarPro	2.88	1.40	1.23	1.84
Unweighted OutPro	$\widehat{S}$	forest	uniform	3.38	3.42	3.80	3.53
All-coordinate OutPro	all	forest	VarPro	2.91	2.94	3.89	3.25
Subspace KNN	$\widehat{S}$	ordinary KNN	VarPro	1.47	2.82	2.30	2.33
Full-space KNN	all	ordinary KNN	uniform	4.37	4.42	3.78	4.19

5.5.1 Results

Figure 8 summarizes the AUC-PR values. Overall, we observe that every non-OPTICS distance attained its best AUC-PR at the largest neighborhood (here equal to $2K_0$), with the Manhattan, Euclidean and Minkowski distances achieving best overall performance. When using OPTICS, AUC-PR similarly improved with K but was essentially flat across `minPts` $\in \{3, 5, 10\}$.

This shows OutPro is stable across its tuning parameters. The default value K_0 worked well, and while larger neighborhoods improved performance, the gains were moderate. The monotone improvement with K is consistent with the design of the method. Because $\mathcal{N}_K(x)$ is constructed from the fitted model's rule regions rather than from a global distance, it already concentrates on training cases that are functionally relevant to x. Enlarging K within this model-derived neighborhood adds useful context without introducing the noise that would arise from expanding a conventional nearest-neighbor search.

5.6 Component ablation

We next isolate the main components of OutPro using the PMLB regression benchmark. The components are the VarPro signal set $\widehat{S}$, the forest neighborhood $\mathcal{N}_K(x^*)$, and the VarPro weights used in the final distance. Here the forest neighborhood means the set of K training points with the largest OutPro proximity scores $W(x_i)$ for the test input x^*, as defined in Sect. 3.3. Ordinary KNN is the set of K training points with the smallest Manhattan distance to x^*, computed using the coordinates and weights specified in Table 5.

To keep the ablation focused on algorithmic components, all settings use the Manhattan distance and the default neighborhood size $K_0 = \min(n/10, 5000)$. Table 5 reports average AUC-PR ranks across the PMLB regression benchmark. Lower ranks indicate better performance.

The results support the integrated construction of OutPro. Full OutPro has the best average rank across the three copula modes, with especially strong performance for `support` and `warp` shifts. Removing VarPro weights while keeping the same selected variables and forest neighborhood reduces performance in all three modes. Removing the subspace restriction by using all coordinates also reduces performance, consistent with signal dilution from nuisance coordinates.

The comparison with KNN shows that the effect of the forest neighborhood depends on the type of shift. Subspace KNN has the best rank for `joint` shifts, but Full OutPro is substantially better for `support` and `warp` shifts and is better overall. Full-space KNN

performs poorly in all settings, indicating that ordinary nearest-neighbor search in the full covariate space is not sufficient.

6 Evaluating lymphadenectomy survival through OOD analysis

We next consider a survival case study from the Worldwide Esophageal Cancer Collaboration (WECC) [74]. The registry includes patients who underwent esophagectomy alone for esophageal cancer, with right censored follow-up on all-cause mortality [75]. Previous WECC analyses used random survival forests (RSF) [72] to study the relation between lymphadenectomy, measured by the number of lymph nodes resected, and five-year survival. Those analyses found that greater lymphadenectomy was generally associated with improved survival, with optimal counts depending on tumor stage and histopathologic type [75]. We revisit this setting to study whether patients receiving more extensive lymphadenectomy become less outlying relative to clinically relevant training populations.

The analysis used $n = 6,142$ adenocarcinoma cases. Covariates included pTNM classification, number of lymph nodes resected, number of positive nodes, histologic grade, tumor location and length, residual cancer status, and patient demographics, giving $d = 35$ variables. In the pTNM system, the prefix "p" denotes pathologic staging from surgical resection specimens, and the T category records depth of tumor invasion from pT1 to pT4. We focus on patients with pT3 or pT4 tumors, combined into a single group.

The upper panel of Fig. 9 gives the RSF out-of-bag five-year survival predictions for pT3–4 adenocarcinoma, stratified by the number of positive nodes, 1–2, 3–6, and ≥ 7. The pattern recovers the clinical signal reported previously: survival improves as more lymph nodes are removed, but the gain is attenuated in patients with heavier nodal involvement.

For OOD scoring, we created 31 train/test scenarios by varying the lymphadenectomy cutoff from 0 to 30 nodes. At each cutoff, the test set contained node-positive pT3–4 patients whose lymph node count met or exceeded the cutoff, and all remaining patients formed the training set. A cutoff of 0 therefore holds out all node-positive pT3–4 cases, a clinically distinct group with deeply invasive disease and nodal involvement. As the cutoff increases, the held-out group is restricted to patients with progressively greater lymphadenectomy. For each scenario, we computed the mean OOD percentile score for the held-out group on a 0–100 scale.

The lower panel of Fig. 9 shows that mean OOD scores decrease as the cutoff increases, indicating that patients with greater lymphadenectomy become less distinct from the corresponding training population. The decline is steepest for patients with 1–2 positive nodes and shallowest for those with ≥ 7 positive nodes. This is consistent with the survival curves in the upper panel, where the benefit of additional lymphadenectomy is less pronounced with heavier nodal involvement. Using the 95th percentile as a threshold for considering a case non-OOD, the required lymphadenectomy count is approximately 3 nodes for the 1–2 and 3–6 positive-node categories, but about 30 nodes for the ≥ 7 category. At a stricter 90th percentile threshold, the required count rises to 5 and 10 nodes for the 1–2 and 3–6 categories, respectively.

Because the surgeon does not know the true nodal status during resection, the OOD analysis supports a more aggressive lymphadenectomy for suspected deeply invasive tumors, on the order of 30 nodes. This finding is consistent with the earlier WECC report, which estimated optimal lymphadenectomy counts for pT3/T4 cancers ranging from 29 to 50 nodes depending on histopathologic type [75].

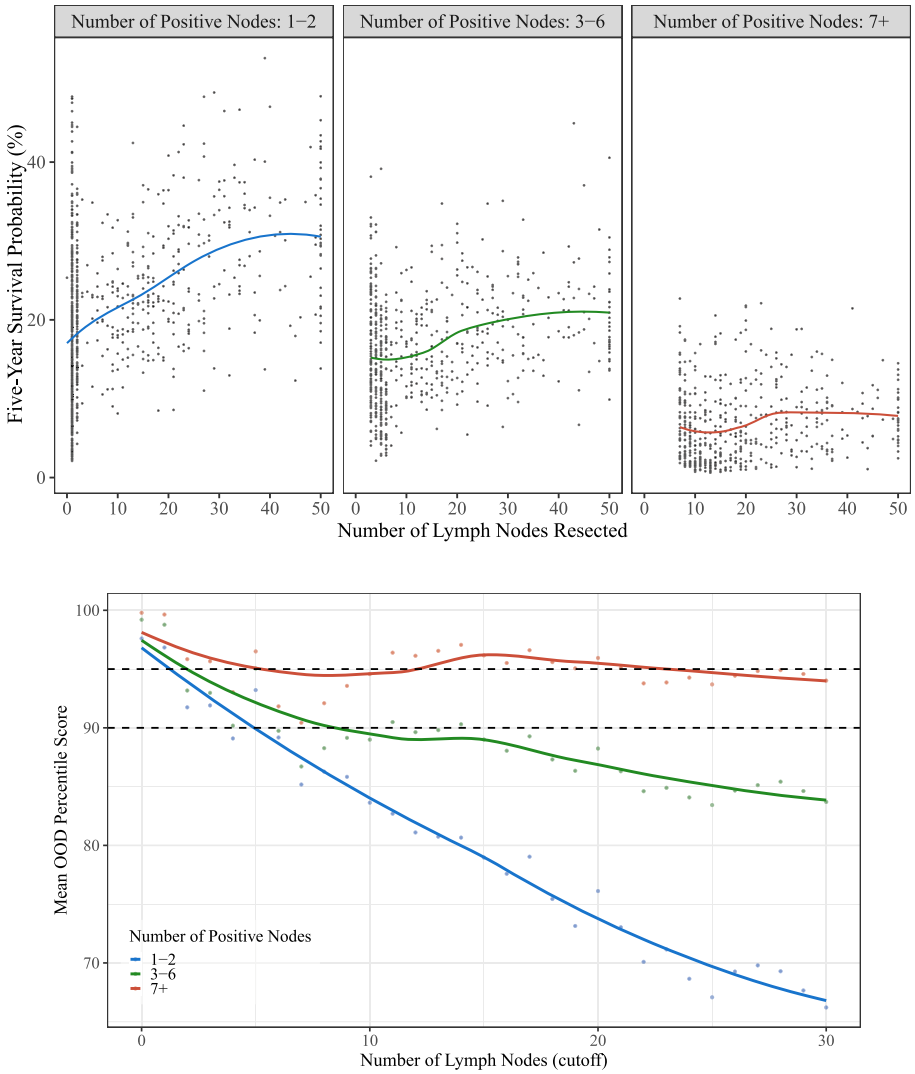

Fig. 9 WECC adenocarcinoma analysis for pT3–4 tumors. Upper panel shows out-of-bag five-year survival predictions from RSF as a function of the number of lymph nodes removed, stratified by the number of positive nodes. Survival generally improves with greater lymphadenectomy, with smaller gains as nodal involvement increases. Lower panel shows mean OOD percentile score as a function of the lymphadenectomy cutoff, for each of 31 hold-out scenarios with cutoffs from 0 to 30. A score of 100 corresponds to the most extreme OOD score under the null training distribution

7 Summary and limitations

We have proposed an integrated model-aware, subspace-aware framework for OOD detection in supervised settings. A key feature of the method is that the OOD score is not constructed from the model's predicted values, residuals, or uncertainty measures, but rather from the rule structure learned by a supervised forest. Those learned rules identify a reference neighborhood

for each test input and select a predictive subspace, after which OOD is measured by distances computed only within that subspace. The fitted model therefore enters the detector through its learned partition of the covariate space rather than through its output, which focuses the score on covariate variation that is relevant for the outcome and separates functionally meaningful departures from shifts occurring mainly in irrelevant coordinates. The framework is general, and the empirical work in this paper focused on regression and survival settings.

The empirical results show that performance depends on the form of the shift. In the Friedman experiments, OutPro performed best for small feature-targeted perturbations, where anomalies are subtle and difficult to distinguish from ordinary test variation. In the PMLB benchmarks, OutPro was best under the `joint` and `support` modes. Under the `warp` mode, however, classical tabular methods such as Isolation Forest, one-class SVM, Local Outlier Factor, and robust kNN density were often among the best procedures. In the high-dimensional microarray studies, OutPro again performed very well under `joint` and remained competitive under `support`, while the `warp` results were more mixed. This pattern is consistent with the structure of `warp`, which creates low-density marginal tail distortion within the observed support and is therefore well matched to density and isolation rules. A sensitivity analysis also suggested that the method is reasonably stable over a broad range of its key tuning parameters, including K (the number of neighbors), with only moderate gains from enlarging the model-derived neighborhood.

The case study on esophageal cancer and lymphadenectomy illustrated the clinical interpretability of the resulting scores. Using pT3-4 adenocarcinoma patients from the WECC dataset, we examined how increasingly aggressive lymphadenectomy moved patients toward or away from anomalous regions relative to the training cohort. The analysis suggested that for suspected deeply invasive tumors, removal of approximately 30 lymph nodes may be a reasonable surgical target when true nodal status is unknown, in line with prior WECC findings.

OutPro was also computationally manageable across all studies considered. Runtimes across the PMLB regression benchmarks and the high-dimensional microarray datasets were well under 30 seconds at their largest, and in the microarray studies were especially short because variable prioritization substantially reduced the working dimension and sample sizes were moderate. The computation benefits from three features of the procedure, namely that the prediction engine is based on random forests, that priority rules can be extracted efficiently from the trained ensemble, and that the final neighborhood scoring step is inexpensive once the predictive subspace has been selected.

Several limitations should be mentioned. First, the empirical results show that no single detector is uniformly best, and within OutPro no single subspace distance dominates across all settings. The product score worked well in lower-dimensional settings, while Manhattan, Euclidean, Minkowski, and Mahalanobis scores were often more stable in higher-dimensional problems, and classical tabular methods were especially competitive under `warp`. A fuller theory for choosing the subspace distance and identifying the settings where each score is preferred remains open. Second, the method depends on estimating a predictive subspace from the fitted model, and errors in that step can affect both the reference neighborhood and the final score. Third, while the copula-based benchmarks provide varied and controlled departures, any synthetic benchmark remains an approximation to the shifts encountered in practice. A fourth limitation concerns pure concept shift. Although the trained forest, signal variables, and reference neighborhoods are learned from labeled training data, the OutPro score for a new input is computed from its covariates. Consequently, two test distributions with the same $\mathbb{P}^*_X$ but different $\mathbb{P}^*_{Y|X}$ induce the same distribution of OutPro scores. The

method is therefore intended for prediction relevant covariate departures, not for detecting conditional shifts that leave the covariate distribution unchanged.

Acknowledgements This work was supported by the National Institute of General Medical Sciences of the National Institutes of Health under Award Number R35 GM139659 and by the National Heart, Lung, and Blood Institute of the National Institutes of Health under Award Number R01 HL164405.

Author Contributions Min Lu: Conceptualization, Methodology, Supervision. Hemant Ishwaran: Conceptualization, Methodology, Writing - Original Draft, Software, Supervision, Writing - Review & Editing, Funding Acquisition.

Data availability Our code is publicly available as a CRAN R-package varPro. All simulated datasets can be reproduced directly from the code. Public benchmark datasets used in this study are available from the PMLB repository, and information on the microarray datasets can be found in the cited references. The esophageal cancer data from the Worldwide Esophageal Cancer Collaboration are not publicly available.

Declarations

Conflicts of Interest The authors declare no conflicts of interest.

References

1. Hendrycks D, and Gimpel K (2016) A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136
2. Salehi M, Mirzaei H, Hendrycks D, Li Y, Rohban M.H, and Sabokrou M (2022) A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *Transactions on Machine Learning Research*, 2022:1–81. ISSN 2835-8856. Article No. 234, published via OpenReview under CC BY 4.0 URL. https://openreview.net/forum?id=aRtjVZvbpK
3. Quiñonero-Candela J, Sugiyama M, Schwaighofer A, Lawrence ND (eds) (2009) Dataset Shift in Machine Learning. MIT Press, Cambridge, MA. https://doi.org/10.7551/mitpress/9780262170055.001.0001
4. Storkey AJ (2009) When training and test sets are different: Characterizing learning transfer. In Quiñonero-Candela J, Sugiyama M, Schwaighofer A, and Lawrence ND, editors, *Dataset Shift in Machine Learning*, pages 3–28. MIT Press, Cambridge, MA, https://doi.org/10.7551/mitpress/9780262170055.003.0001
5. Shimodaira H (2000) Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of Statistical Planning and Inference 90(2):227–244. https://doi.org/10.1016/S0378-3758(00)00115-4
6. Sugiyama M, Krauledat M, Müller KR (2007) Covariate shift adaptation by importance weighted cross validation. J Mach Learn Res 8:985–1005
7. Moreno-Torres JG, Raeder T, Alaiz-Rodríguez R, Chawla NV, Herrera F (2012) A unifying view on dataset shift in classification. Pattern Recogn 45(1):521–530. https://doi.org/10.1016/j.patcog.2011.06.019
8. Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW (2010) A theory of learning from different domains. *Machine Learning*, 79(1–2):151–175, https://doi.org/10.1007/s10994-009-5152-4
9. Wu Y, Li T, Cheng X, Yang J, Huang X (2024) Low-dimensional gradient helps out-of-distribution detection. IEEE Trans Pattern Anal Mach Intell 46(12):11378–11391
10. Rabanser S, Günnemann S, and Lipton ZC (2019) Failing loudly: An empirical study of methods for detecting dataset shift. In *Advances in Neural Information Processing Systems*, volume 32

11. Yang J, Zhou K, Li Y, Liu Z (2024) Generalized out-of-distribution detection: A survey. Int J Comput Vision 132(12):5635–5662
12. Tamang L, Bouadjenek MR, Dazeley R, Aryal S (2025) Handling out-of-distribution data: A survey. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2025.3592614. Early Access
13. Lu S, Wang Y, Sheng L, He L, Zheng A, Liang J (2025) Out-of-distribution detection: A task-oriented survey of recent advances. ACM Comput Surv 58(2):1–39
14. Zhang K, Liu S, Wang S, Shi W, Chen C, Li P, Li S, Li J, and Ding K (2024) A survey of deep graph learning under distribution shifts: from graph out-of-distribution generalization to adaptation. arXiv preprint arXiv:2410.19265,
15. Li H, Wang X, Zhang Z, Zhu W (2025) Out-of-distribution generalization on graphs: A survey. IEEE Trans Pattern Anal Mach Intell
16. Wu X, Teng F, Li X, Zhang J, Li T, Duan Q (2025) Out-of-distribution generalization in time series: A survey. arXiv preprint arXiv:2503.13868,
17. Liang S, Li Y, Srikant R (2018) Enhancing the reliability of out-of-distribution image detection in neural networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*,. URL https://arxiv.org/abs/1706.02690
18. Liu W, Wang X, Owens J, Li Y (2020) Energy-based out-of-distribution detection. Adv Neural Inf Process Syst 33:21464–21475
19. Lee K, Lee H, Lee K, Shin J (2018) A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 7167–7177. URL https://papers.nips.cc/paper/2018/hash/a19744e268754fb0148b017647355dd3-Abstract.html
20. Sun Y, Ming Y, Zhu X, Li Y (2022) Out-of-distribution detection with deep nearest neighbors. In *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 20827–20840,
21. Ming Y, Fan Y, Li Y (2022) POEM: Out-of-distribution detection with posterior sampling. In *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 15650–15665,
22. Zhang Y, Lu J, Peng B, Fang Z, Cheung Y-M (2024) Learning to shape in-distribution feature space for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 37, pages 49384–49402,
23. Chen Q, Li K, Chen Z, Maul T, Yin J (2024) Exploring feature sparsity for out-of-distribution detection. Sci Rep 14(1):28444
24. Zöngür B, Hesse R, and Roth S (2025) Activation subspaces for out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3509–3519
25. Zhu Q, He Y (2024) Out-of-distribution detection based on subspace projection of high-dimensional features output by the last convolutional layer. arXiv preprint arXiv:2405.01662
26. Fang K, Tao Q, Lv K, He M, Huang X, Yang J (2024) Kernel PCA for out-of-distribution detection. In Advances in Neural Information Processing Systems 37:134317–134344
27. Lakshminarayanan B, Pritzel A, and Blundell C (2017) Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, URL https://arxiv.org/abs/1612.01474
28. Gal Y, Ghahramani Z (2016) Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1050–1059, URL https://arxiv.org/abs/1506.02142
29. Amini A, Schwarting W, Soleimany A, and Rus D (2020) Deep evidential regression. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, URL https://arxiv.org/abs/1910.02600
30. Malinin A, Gales M (2018) Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, volume 31
31. Pequignot Y, Alain M, Dallaire P, Yeganehparast A, Germain P, Desharnais J, Laviolette F (2023) Out-of-distribution detection for regression tasks: parameter versus predictor entropy, URL https://arxiv.org/abs/2010.12995
32. Gramacy RB (2016) laGP: Large-scale spatial modeling via local approximate Gaussian processes in R. Journal of Statistical Software 72(1):1–46. https://doi.org/10.18637/jss.v072.i01
33. Wilson AG, Hu Z, Salakhutdinov R, Xing EP (2016) Deep kernel learning. In Gretton A. and Robert C.C, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 370–378, Cadiz, Spain, PMLR. URL https://proceedings.mlr.press/v51/wilson16.html
34. van Amersfoort J, Smith L, Jesson A, Key O, Gal Y (2021) On feature collapse and deep kernel learning for single forward pass uncertainty. arXiv preprint arXiv:2102.11409,

35. Knorr EM, Ng RT (1998) Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24th International Conference on Very Large Data Bases*, pages 392–403
36. Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 427–438. ACM
37. Breunig MM, Kriegel HP, Ng RT, Sander J (2000) LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp 93–104. ACM
38. Scholkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC (2001) Estimating the support of a high-dimensional distribution. Neural Comput 13(7):1443–1471
39. Liu F.T, Ting K.M, and Zhou ZH (2008) Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE,
40. Tosaki T, Uchino E, Kojima R, Mineharu Y, Okamoto Y, Arita M, Miyai N, Mikami T, Murashita K et al (2025) Out-of-distribution reject option method for dataset shift problem in early disease onset prediction. Sci Rep 15(1):19240
41. Azizmalayeri M, Abu-Hanna A, Cina G (2025) Unmasking the chameleons: A benchmark for out-of-distribution detection in medical tabular data. Int J Med Informatics 195:105762
42. Zadorozhny K, Thoral P, Elbers P, Cinà G (2022) Out-of-distribution detection for medical applications: Guidelines for practical evaluation. In *Multimodal AI in Healthcare: A Paradigm Shift in Health Intelligence*, pages 137–153. Springer
43. Zamzmi G, Venkatesh K, Nelson B, Prathapan S, Yi P, Sahiner B, Delfino JG (2025) Out-of-distribution detection and radiological data monitoring using statistical process control. *Journal of Imaging Informatics in Medicine*, 38(2):997–1015
44. Angelopoulos AN, Bates S (2021) A gentle introduction to conformal prediction and distribution-free uncertainty quantification. J Mach Learn Res 22(55):1–69
45. Bates S, Candès E, Lei L, Romano Y, Sesia M (2023) Testing for outliers with conformal p-values. Ann Stat 51(1):149–178
46. Liang Z, Sesia M, Sun W (2024) Integrative conformal p-values for powerful out-of-distribution testing with labeled outliers. J Roy Stat Soc B 86(3):671–700
47. Magesh A, Wang Y et al (2023) Principled out-of-distribution detection via multiple testing. J Mach Learn Res 24(378):1–35
48. Novello P, Dalmau J, and Andeol L. Out-of-distribution detection should use conformal prediction (and vice-versa?). arXiv preprint arXiv:2403.11532, 2024
49. Breiman L (2001) Random forests. Mach Learn 45:5–32
50. Lu M, Ishwaran H (2024) Model-independent variable selection via the rule-based variable priority. arXiv, 2409.09003, URL https://arxiv.org/abs/2409.09003
51. Ishwaran H (2025) Multivariate Statistics: Classical Foundations and Modern Machine Learning, 1st edn. Chapman and Hall/CRC, Boca Raton, FL
52. Lu M, Ishwaran H (2025) Individual variable priority: a model-independent local gradient method for variable importance. Artif Intell Rev 58(12):407
53. Zhou L, Lu M, Ishwaran H (2026) Variable priority for unsupervised variable selection. Pattern Recogn 172:112727. https://doi.org/10.1016/j.patcog.2025.112727
54. Ankerst M, Breunig M.M, Kriegel H.P, Sander J (1999) Optics: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pages 49–60. ACM Press, https://doi.org/10.1145/304181.304187. URL https://doi.org/10.1145/304181.304187
55. Hahsler M, Piekenbrock M, Doran D (2019) dbscan: Fast density-based clustering with R. *Journal of Statistical Software*, 91(1):1–30, https://doi.org/10.18637/jss.v091.i01. URL https://doi.org/10.18637/jss.v091.i01
56. Chollet F (2015) Keras, https://github.com/keras-team/keras
57. Gramacy RB *laGP: Local Approximate Gaussian Process Regression*, (2021) URL https://cran.r-project.org/package=laGP. R package version 1.5-7
58. He W, Li X, Liang Y, Song L, Li S (2018) Softmax response for out-of-distribution detection. In *Neural Information Processing Systems (NeurIPS) Workshop on Out-of-Distribution Generalization*, Demonstrates sensitivity (gradient) for OOD
59. Pleiss G, Souza A, Kim J, Li B, Weinberger KQ (2020) Neural network out-of-distribution detection for regression tasks. URL https://openreview.net/forum?id=ryxsUySFwr. ICLR Submission (preprint)
60. Friedman JH (1991) Multivariate adaptive regression splines. Ann Stat 19(1):1–67. https://doi.org/10.1214/aos/1176347963

61. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232
62. Sklar A (1973) Random variables, joint distribution functions, and copulas. Kybernetika 9(6):449–460
63. Joe H (2014) Dependence Modeling with Copulas. CRC Press,
64. Nelsen RB (2006) An Introduction to Copulas, 2nd edn. Springer Series in Statistics. Springer
65. Olson RS, La Cava W, Mustahsan Z, Varik G, Moore JH (2017) PMLB: A large benchmark suite for machine learning evaluation and comparison. BioData Mining 10(1):36. https://doi.org/10.1186/s13040-017-0154-4
66. Palaniappan L, Lengerich B.J, Olson RS, Moore JH (2020) *pmlbr: R Interface to the Penn Machine Learning Benchmark (PMLB)*, URL https://CRAN.R-project.org/package=pmlbr. R package version 0.2.1
67. Rosenwald A, Wright G, Chan WC, Connors JM, Campo E, Fisher RI, Gascoyne RD, Muller-Hermelink HK, Smeland EB, Staudt LM (2002) The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. N Engl J Med 346:1937–1947
68. van't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AA, Mao M, Peterse HL, van der Kooy K, Marton MJ, Witteveen AT, et al (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536
69. Beer DG, Kardia SL, Huang CC, Giordano TJ, Levin AM, Misek DE, Lin L, Chen G, Gharib TG, Thomas DG et al (2002) Gene-expression profiles predict survival of patients with lung adenocarcinoma. Nat Med 8:816–824
70. Bullinger L, Döhner K, Bair E, Fröhling S, Schlenk RF, Tibshirani R, Döhner H, Pollack JR (2004) Use of gene-expression profiling to identify prognostic subclasses in adult acute myeloid leukemia. *New England Journal of Medicine*, 350:1605–1616
71. Rosenwald A, Wright G, Wiestner A, Chan WC, Connors JM, Campo E, Gascoyne RD, Grogan TM, Muller-Hermelink HK, Smeland EB et al (2003) The proliferation gene expression signature is a quantitative integrator of oncogenic events that predicts survival in mantle cell lymphoma. Cancer Cell 3:185–197
72. Ishwaran H, Kogalur U.B, Blackstone E.H, and Lauer M.S. Random survival forests. *The Annals of Applied Statistics*, 2(3):841–860, 2008
73. Bair E, Tibshirani R (2004) Semi-supervised methods to predict patient survival from gene expression data. PLoS Biol 2:511–522
74. Rice T, Rusch V, Apperson-Hansen C et al (2009) Worldwide esophageal cancer collaboration. Dis Esophagus 22:1–8
75. Rizk NP, Ishwaran H, Rice TW, Chen LQ, Schipper PH, Kesler KA, Law S, Lerut TE, Reed CE, Salo JA, Scott WJ, Hofstetter WL, Watson TJ, Allen MS, Rusch VW, and Blackstone EH. Optimum lymphadenectomy for esophageal cancer. *Annals of Surgery*, 251(1):46–50, (2010) https://doi.org/10.1097/SLA.0b013e3181b2f6ee

Min Lu is a Research Associate Professor in the Division of Biostatistics, Department of Public Health Sciences, at the University of Miami Miller School of Medicine. She earned a B.S. in Economics from Dalian University of Technology and a Ph.D. in Biostatistics from the University of Miami. Her research focuses on machine learning methods for public health and medicine, particularly variable selection, causal inference, and high-dimensional modeling. She has co-developed open-source R packages for random forests and model-independent variable selection, with applications in oncology, cardiothoracic surgery, genomics, and epidemiology.

Hemant Ishwaran is a Professor in the Division of Biostatistics, Department of Public Health Sciences, at the University of Miami Miller School of Medicine. He holds degrees in mathematics and statistics from the University of Toronto, an M.Sc. from the University of Oxford, and a Ph.D. from Yale University. His research focuses on machine learning methods for public health and medicine, particularly high-dimensional modeling and survival analysis. He developed random survival forests and several open-source R packages. His work has contributed to cancer staging and clinical decision tools in cardiothoracic surgery and oncology.