The Need for Unsupervised Outlier Model Selection: A Review and Evaluation of Internal Evaluation Strategies

Martin Q. Ma†, Yue Zhao†, Xiaorong Zhang, and Leman Akoglu
Carnegie Mellon University, Pittsburgh, PA, USA
qianlim@cs.cmu.edu, zhaoy@cmu.edu, xiaoronz@alumni.cmu.edu, lakoglu@andrew.cmu.edu

ABSTRACT

Given an *unsupervised* outlier detection task, how should one select i) a detection algorithm, and ii) associated hyperparameter values (jointly called a model)? Effective outlier model selection is essential as different algorithms may work well for varying detection tasks, and moreover their performance can be quite sensitive to the values of the hyperparameters (HPs). On the other hand, unsupervised model selection is notoriously difficult, in the absence of hold-out validation data with ground-truth labels. Therefore, the problem is vastly understudied in the outlier mining literature. There exists a body of work that propose *internal model evaluation strategies* for selecting a model. These so-called internal strategies solely rely on the input data (without labels) and the output (outlier scores) of the candidate models. In this paper, we first survey internal model evaluation strategies including both those proposed specifically for outlier detection, as well as those that can be adapted from the unsupervised deep representation learning literature. Then, we investigate their effectiveness empirically in comparison to simple baselines such as random selection and the popular state-of-the-art detector Isolation Forest (iForest) with *default* HPs. To this end, we set up (and open-source) a large testbed with 39 detection tasks and 297 candidate models comprised of 8 different detectors and various HP configurations. We evaluate internal strategies from 7 different families on their ability to discriminate between models w.r.t. detection performance, without using any labels. Our study reports a striking finding, that *none of the existing and adapted strategies would be practically useful*: stand-alone ones are not significantly different from random, and consensus-based ones do not outperform iForest (w/ default HPs) while being more expensive (as *all* candidate models need to be trained for evaluation). Our survey stresses the importance of and the standing need for effective unsupervised outlier model selection, and acts as a call for future work on the problem.

1. INTRODUCTION

Model selection aims to select a model from a set of candidate models for a task, given data. We consider the model selection problem for the *unsupervised* outlier detection (UOD) task. Specifically, given a dataset for UOD, how can we identify – *without using any labels* – which outlier model (a detection algorithm and the value(s) of its hyperparameter(s)) performs better than the others on the input dataset? Importantly, note that as the outlier detection task is unsupervised, so is the model selection task. That is, an outlier model is to be selected without being able to validate any candidate models on hold-out labeled data.

Motivation and Challenges. The notion of a universally "best" outlier model does not exist; rather the best-performing model depends on the given data. On the other hand, model selection is a nontrivial one, provided there are numerous outlier detection algorithms based on a variety of approaches: distance-based [17; 36], density-based [4; 12; 43], angle-based [19], ensemble-based [2; 27; 35], most recently deep neural network (NN) based [7; 9; 40; 45], and so on. To add to this "choice paralysis", most models are sensitive to their choice of hyperparameters (HPs) with significant variation in performance [13], even more so for *deep* neural network (NN) based outlier models that have a long list of HPs [9]. Unsupervised model selection will likely be an increasingly pressing problem for deep detectors, as their complexity and expressiveness grow. Recent work hold out some labeled validation data for tuning such deep outlier models [22; 23; 40], which however is not feasible for fully unsupervised settings. These factors make outlier model selection a problem of utmost importance.

Despite its importance, the problem of Unsupervised Outlier Model Selection (UOMS hereafter) is a notoriously challenging one. Mainly, the absence of validation data with labels makes the problem hard. Moreover, there does not exist a universal or well-accepted objective criterion (i.e. loss function) for outlier detection, which makes model comparison infeasible. Besides its unsupervised nature, the search space for UOMS can be quite large with the arrival/popularity of deep NN based models with many HPs.

Existing work. Perhaps due to these challenges, UOMS remains a vastly understudied area. Most work in the outlier mining literature focus on designing new detection algorithms, such as those for unique settings: streaming [14; 28; 42], contextual [24; 31], human-in-the-loop [8; 21] detection, etc. There exist a small body of work, specifically addressing UOMS, that proposes *internal* (i.e. unsupervised) model evaluation strategies to assess the quality of a model and its output. These are called internal strategies as they use heuristic measures that solely make use of the input data and/or output outlier scores. To our knowledge, there are only three such techniques (in chronological order) [30; 11; 34]. However, they employ their proposed strategies to select only among 2-3 detectors on 8-12 real-world datasets. More

†Equal contribution.

problematically, they do not systematically compare to one another, nor do they use the same datasets. This makes it difficult to fully understand the strengths and limitations of these existing methods, and ultimately the extent to which progress has been made on this subject.

More recently, we have designed a series of new approaches for UOMS leveraging two key concepts; meta-learning [51; 48; 49] and hyper-ensembles [9]. Notably, the former works utilize internal evaluation measures, the focus of this survey. The idea is to boost the relatively weak internal model performance signals from these heuristic measures via meta-learning from historical tasks that have labels. As such, any future work on designing new internal model evaluation strategies and improving their effectiveness and speed would directly feed into and advantage these meta-learning based UOMS approaches. (See Sec. 4 for detailed related work.)

Our survey. In this work, our goal is to survey internal model evaluation strategies, and systematically evaluate and compare their effectiveness. To this end, we first bring under one umbrella the aforementioned three existing UOMS methods, adapt two state-of-the-art unsupervised model selection techniques originally proposed for deep representation learning [10; 25], and design two new internal model selection methods inspired by various consensus algorithms. We put them to test on a large testbed against simple baselines, including random model selection as well as the popular isolation Forest (iForest) [27], with default HPs. To our knowledge, this is the first work to systematically review and evaluate the internal evaluation strategies toward unsupervised model selection for outlier detection. We summarize the contributions and findings of this paper as follows.

- **Unified Comparison:** We identify (to our knowledge) all existing internal model evaluation strategies for UOMS. For the first time, we systematically compare them on their ability to discriminate between models w.r.t. detection performance, as well as w.r.t. running time, on the same testbed.
- **Large-scale Evaluation:** Our testbed consists of 8 state-of-the-art detectors, each configured by a comprehensive list of hyperparameter settings, yielding a candidate pool of 297 models.We perform the model selection task on 39 independent real-world datasets from two different public repositories. We compare different strategies through paired statistical tests to identify significant differences, if any. We find that all three existing strategies specifically designed for UOMS are ill-suited. Alarmingly, none of them is significantly different from random selection (!)
- **New UOMS Techniques:** All three existing methods specifically designed for UOMS are *stand-alone*; evaluating each model individually, independent of others. In addition to those, we repurpose four *consensus-based* algorithms from other areas for UOMS; utilizing the agreements among the models in the pool. We find that consensus-based methods are more competitive than stand-alone ones, and all of them achieve significantly better performance than random. However, they are not different from iForest (the best detector in our pool), thus, would not be employed (on a pool) over training a single (iForest) model.
- **Open-source Testbed:** We expect that UOMS will continue to be a pressing problem, especially with the advent of deep detection models with many hyperparameters. Our large-scale analysis reveals that there is ample room for progress on this problem, and serves as a call for future work. At the same time, our results shed light onto the strengths and limitations of different approaches that motivate future directions.

Reproducibility and Future Work. To foster progress on this key problem, we open-source all datasets, our trained model pool, and implementations of all the internal model evaluation strategies at `https://github.com/yzhao062/uoms`.

2. PRELIMINARIES & THE PROBLEM

Model selection concerns with picking a model from a pool of candidate models. Let $\mathcal{M} = \{M_i\}_{i=1}^{N}$ denote a pre-specified pool of N models. Here each model M_i is a {`detector`, `HPconfiguration`} pair, where `detector` is a certain outlier detection algorithm (e.g. Local Outlier Factor (LOF) [5]) and `HPconfiguration` is a certain setting of the values for its hyperparameter(s) (e.g. for LOF, value of `n_neighbors`: number of nearest neighbors to consider, and function of choice for `distance` computation).

In this study, $\mathcal{M}$ is composed by pairing 8 popular outlier detection algorithms to distinct hyperparameter choices, comprising a total of $N = 297$ models, as listed in Table 1. All models are trained based on the Python Outlier Detection Toolbox (PyOD) [50] on each dataset.

Let $\mathcal{D} = \{D_t\}_{t=1}^{T}$ denote the set of outlier detection datasets (i.e. tasks), where $D_t = \{\mathbf{x}_j^{(t)}\}_{j=1}^{n_t}$, $n_t = |D_t|$ is the number of samples and o_t is the true number of ground-truth outliers in D_t. We denote by $\mathbf{s}_i^{(t)} \in \mathbb{R}^{n_t}$ the list of outlier scores output by model M_i when employed (i.e. trained[1]) on D_t, and $s_{ij}^{(t)} \in \mathbb{R}$ to depict individual sample j's score. We omit the superscript when it is clear from context. W.l.o.g. the higher the s_{ij} is, the more anomalous is j w.r.t. M_i.

PROBLEM 1 (UNSUPERVISED OUTLIER MODEL SELECTION). (UOMS) *The model selection problem for unsupervised outlier detection can be stated as follows.*

Given	an unsupervised detection task $D = \{\mathbf{x}_j\}_{j=1}^{n}$, all models in $\mathcal{M}$ trained on D with corresponding output scores $\{\mathbf{s}_i\}_{i=1}^{N}$;
Select	a model $M' \in \mathcal{M}$,
such that	$\mathbf{s}'$ yields good detection performance.

Note that the detection performance is to be quantified *post* model selection, where ground-truth labels are used only for evaluation (and **not** for model training or model selection).

In this work, we study 7 different families of internal strategies (See Table 2): (1) three techniques that were proposed to directly address the UOMS problem, (2) two unsupervised model selection techniques adopted from deep learning, and (3) two others that are not originally designed for model selection that we adapt to UOMS.

To compare their effectiveness systematically, we construct a large testbed of $T = 39$ real-world outlier detection datasets from two different repositories (See Sec. 5.1). That is, we perform UOMS using each technique 39 times, to select one model from the pool of 297. Given that the datasets are independent, a large testbed enables paired statistical tests that conclusively identify significant differences between these techniques and various simple baselines.

[1]Note that as we consider unsupervised outlier detection, model "training" does not involve any ground-truth labels.

Table 1: Outlier Detection Models; see hyperparameter definitions from PyOD

Detection algorithm	Hyperparameter 1	Hyperparameter 2	Total
LOF [4]	n_neighbors: $[1, 5, 10, 15, 20, 25, 50, 60, 70, 80, 90, 100]$	distance: ['manhattan', 'euclidean', 'minkowski']	36
kNN [36]	n_neighbors: $[1, 5, 10, 15, 20, 25, 50, 60, 70, 80, 90, 100]$	method: ['largest', 'mean', 'median']	36
OCSVM [41]	nu (train error tol): $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$	kernel: ['linear', 'poly', 'rbf', 'sigmoid']	36
COF [43]	n_neighbors: $[3, 5, 10, 15, 20, 25, 50]$	N/A	7
ABOD [19]	n_neighbors: $[3, 5, 10, 15, 20, 25, 50]$	N/A	7
iForest [27]	n_estimators: $[10, 20, 30, 40, 50, 75, 100, 150, 200]$	max_features: $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$	81
HBOS [12]	n_histograms: $[5, 10, 20, 30, 40, 50, 75, 100]$	tolerance: $[0.1, 0.2, 0.3, 0.4, 0.5]$	40
LODA [35]	n_bins: $[10, 20, 30, 40, 50, 75, 100, 150, 200]$	n_random_cuts: $[5, 10, 15, 20, 25, 30]$	54
			297

3. REVIEW OF INTERNAL MODEL EVALUATION STRATEGIES

Internal strategies evaluate the goodness of a model without using any external information, especially with no access to ground-truth labels. The internal information being used is solely limited to (i) the input samples (feature values only), (ii) the trained models in the candidate pool and the outlier scores as output by these trained models. The common thread among all internal model evaluation strategies in this study is an estimated heuristic *internal measure* of "model goodness". Model selection is then addressed by top-1 selection: i.e. picking the model with the highest value of the respective measure.

We categorize the 7 strategies we studied into two, depending on how they estimate their internal measure: (1) **stand-alone** and (2) **consensus-based**. (See Table 2.) Stand-alone strategies solely rely on each model and its output individually, independent of other models. All three existing methods proposed specifically for UOMS fall into this category. On the other hand, consensus-based strategies leverage agreement between the models in the pool and hence utilize candidate models collectively. Four strategies we adopt and adapt[2] from other areas all fall into this latter category.

In the following we provide a short description of each strategy (and refer to the original articles for full details). We also remark on the computational complexity of some methods as they demand considerable running time. Ideal is to have a lightweight and effective selection method with low overhead incurred on top of model training. In the experiments, we compare these methods w.r.t. their selection performance as well as running time.

3.1 Stand-alone Internal Evaluation (Existing)

3.1.1 IREOS

The first known index proposed for the internal evaluation of outlier detection results is by Marques et al., called Internal, Relative Evaluation of Outlier Solutions (IREOS) [30]. While their initial index is designed only for binary solutions (referred to as "top-n" detection), their recent work [29] generalized to numeric outlier scorings, which is the setting considered in this study.

Their intuition is that an outlier should be more easily separated (discriminated) from other samples than an inlier.

[2]We *adopt* two strategies originally proposed for unsupervised model selection for deep representation learning "as is", and *adapt* two techniques (from information retrieval and ensemble learning) by repurposing them to UOMS problem with small modifications.

Table 2: Overview of UOMS methods studied in this survey.

Method	Type	Based on	Strategy
XB,RS,... [34]	Stand-alone	Outlier scores	Cluster quality
EM, MV [11]	Stand-alone	Outlier scores	Level sets
IREOS [30]	Stand-alone	O. scores + Input	Separability
UDR [10]	Consensus	Outlier scores	One-shot
MC [25], MC_S	Consensus	Outlier scores	One-shot
HITS [16]	Consensus	Outlier scores	Iterative
ENS [52]	Consensus	Outlier scores	Iterative

Then, a model is "good" the more it identifies as outlier those samples with a large degree of separability. They propose to assess the separability of each individual sample using a maximum-margin classifier (and specifically use nonlinear SVMs).[3] The IREOS score of a model M_i on a given dataset is computed as

$$\text{IREOS}(\mathbf{s}_i) = \frac{1}{n_\gamma} \sum_{l=1}^{n_\gamma} \frac{\sum_{j=1}^{n} p(\mathbf{x}_j, \gamma_l) w_{ij}}{\sum_{j=1}^{n} w_{ij}} \qquad (1)$$

where $p(\mathbf{x}_j, \gamma_l)$ is the separability of sample j as estimated by a nonlinear SVM with kernel bandwidth (a hyper-parameter) γ_l, and n_γ is the number of different bandwidth values used from the interval $[0, \gamma_{\max}]$.[4] They convert outlier scores $\{s_{ij}\}_{j=1}^{n}$ to probability weights $\{w_{ij}\}_{j=1}^{n}$ using the approach by [18] to push inlier scores toward zero so that they do not in aggregate dominate the weighted sum. IREOS tends to give high scores to those models whose outlier scores correlate well with the separability scores by a nonlinear SVM.

Computationally, IREOS is quite demanding as it requires training of a nonlinear classifier *per sample*. Their source code[10] provides ways to approximate IREOS scores, mainly estimating separability via nearest neighbor distances, which however are also expensive to compute.

3.1.2 Mass-Volume (MV) and Excess-Mass (EM)

Goix [11] proposed using statistical tools, namely MV and EM curves, to measure the quality of a scoring function. Formally, a scoring function $s : \mathbb{R}^d \mapsto \mathbb{R}_+$ is any measurable function integrable w.r.t. the Lebesgue measure $\text{Leb}(\cdot)$, whose level sets are estimates of the level sets of the density. Outliers are assumed to occur in the tail of the score distribution as produced by a scoring function, where the *lower* $s(\mathbf{x})$ is, the more abnormal is $\mathbf{x}$.

[3]Note that collective outliers (forming micro-clusters, or clumps) do not have high separability. IREOS accounts for this effectively, provided a user-specified `clump_size`. For details, we refer to the original articles.

[4]They use heuristics to automatically set $\gamma_{\max}$ in code.

Given a scoring function $s(\cdot)$ (in our context, an outlier model), the MV measure is defined as follows.

$$\widehat{MV}_s(\alpha) = \inf_{u \geq 0} \text{Leb}(s \geq u)\ s.t.\ \mathbb{P}_n(s(\mathbf{X}) \geq u) \geq \alpha \qquad (2)$$

where $\alpha \in (0, 1)$, and $\mathbb{P}_n$ is the empirical distribution; $\mathbb{P}_n(s \geq v) = \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}_{s(\mathbf{x}_j)>v}$.

For univariate real numbers, Leb$(\cdot)$ measures the length of the given interval. Let $s_{\max}$ denote the largest score produced by $s(\cdot)$. Then, empirically Leb$(s \geq u)$ is equal to the length $|s_{\max} - u|$. Given α, the u that minimizes the Lebesgue measure Leb$(s \geq u)$ in Eq. (2) would be equal to the outlier score at the $(1-\alpha)$-*th* quantile, i.e. $u = CCDF_s^{-1}(\alpha)$. Then, $|s_{\max} - u|$ would give the length of the range of scores for α fraction of the samples with scores larger than u. In their work, they consider $\alpha \in (0.9, 0.999)$.[5,6] As they assume a *lower* score is more anomalous, the Lebesgue measure quantifies the length of the interval of scores for the inliers. The smaller MV is, the better the scoring function is deemed to be. Intuitively, then, MV measures the clusteredness of inlier scores (or the compactness of high-density level sets).

The EM measure is quite similar, and is defined as

$$\widehat{EM}_s(t) = \sup_{u \geq 0}\ \mathbb{P}_n(s(\mathbf{X}) \geq u) - t\text{Leb}(s \geq u) \qquad (3)$$

for $t > 0$. Similarly, they consider $t \in [0, \widehat{EM}_s^{-1}(0.9)]$ with $\widehat{EM}_s^{-1}(0.9) := \inf\{t \geq 0, \widehat{EM}_s(t) \leq 0.9\}$.

Intuitively, EM would identify as small a u value as possible (so as to maximize the density mass in the first term) such that the scores larger than or equal to u are as clustered as possible (so as to minimize the Lebesgue measure in the second term). Again, the more clustered are the scores of the bulk of the samples (i.e. inliers), the larger EM gets, and the better the scoring function is deemed to be.

3.1.3 Clustering validation metrics

[34] point out that a drawback of IREOS, besides computational demand, is its dependence on classification – which itself introduces a model selection problem – since the results may depend on the selected classification algorithm and its hyper-parameter settings.[7] Despite citing IREOS, they do not compare in experiments.

Their key proposal is to apply internal validation measures for clustering algorithms to outlier detection. As clustering aims to ensure samples within each cluster are similar and different from samples in other clusters, these measures are mainly based on compactness (capturing within-cluster similarity) and/or separation (reflecting inter-cluster distance).

To that end, we split the outlier scores by a given model under evaluation for dataset D_t into two clusters, denoted C_o and C_i, respectively consisting of the highest o_t scores and the rest. According to those measures, an outlier model is "good" the more separated these two sets of scores are and/or the more clustered the scores within each set are.

[5]Given fraction of outliers is bounded to 10% maximum.

[6]Area under the MV-curve is estimated as the sum of empirical MV values by Eq. (2) for discretized values of α.

[7]Another paper [33] by the same authors proposed a classification based internal evaluation method, similar to IREOS. Their experiments show that the current internal measures do comparably well or better with less computational overhead, hence we omit [33] from this study.

In their study, they compared 10 different existing clustering quality measures, such as the Silhouette index [39], Xie-Beni index [46], etc. (See others in the original article.) One of the well-performing ones in our experiments, namely Xie-Beni index of a model M_i, denoted XB_i, is defined as follows.

$$\text{XB}_i = \frac{\sum_{j \in C_o} d^2(s_{ij}, c_o) + \sum_{j' \in C_i} d^2(s_{ij'}, c_i)}{n_t\ d^2(c_o, c_i)} \qquad (4)$$

where $c_o = \sum_{j \in C_o} s_{ij}/o_t$ and $c_i = \sum_{j' \in C_i} s_{ij'}/(n_t - o_t)$ depict the cluster centers and $d(\cdot, \cdot)$ is the Euclidean distance. This index can be interpreted as the ratio of the intra-cluster compactness to the inter-cluster separation.

Clustering quality based measures are typically easy to compute; most of them being linear in the number of samples.

3.2 Consensus-based Internal Evaluation (Repurposed)

3.2.1 UDR

The first consensus-based approach, namely Unsupervised Disentanglement Ranking (UDR), is adopted from deep learning and is "the first method for unsupervised model selection for variational disentangled representation learning" [10]. Each model in their case corresponds to a {`HPconfiguration`, `seed`} pair. Reciting Tolstoy who wrote "Happy families are all alike; every unhappy family is unhappy in its own way.", their main hypothesis is that a model with a good hyperparameter (HP) setting will produce *similar results* under different random initializations (i.e. seeds) whereas for a poor HP setting, results based on different random seeds will look arbitrarily different.

In a nutshell, UDR follows 4 steps: (1) Train $N = H \times S$ models, where H and S are the number of hyperparameter settings and random seeds, respectively. (2) For each model M_i, randomly sample (without replacement) $P \leq S$ other models with the *same* HP as M_i, but *different* seeds. (3) Perform P pairwise comparisons between M_i and the models sampled in Step 2 for M_i. (4) Aggregate pairwise similarity scores (denoted $UDR_{ii'}$) as $UDR_i = \text{median}_{i'}\ UDR_{ii'}$, for $i = 1, \ldots, N$. Finally, they pick the model (among N) with the largest UDR_i. Intuitively, UDR selects a model with an HP setting that yields stable or *consistent* results across various seeds.

Notice that adopting UDR for the UOMS task is trivial by making the analogy between {`HPconfiguration`, `seed`} and {`detector`, `HPconfiguration`}. While trivially applied, one may question whether the implied hypothesis (that a good detector has consistent results across different HP settings) holds true for outlier models, since one of the key reasons for UOMS in the first place is that most detectors are sensitive to their HP settings [13].

The key part of UDR is how pairwise model comparisons are done in Step 3. We measure the output *ranking similarity* of the samples by two models, based on three well-known measures from information retrieval [26] (See Sec. 5.1).

3.2.2 MC

A follow-up work to UDR proposed ModelCentrality (MC), which is another consensus-based strategy for what they call "self-supervised" model selection for disentangling GANs [25]. Their premise is similar, that "well-disentangled models should be close to the optimal model, and hence also close

to each other". Provided the similarity $B_{ii'}$ between two models M_i and $M_{i'}$ can be computed, ModelCentrality of M_i is written as $MC_i = \frac{1}{N-1}\sum_{i' \neq i} B_{ii'}$. They then select the model with the largest MC_i, which coincides with the medoid in the pool of models – hence the name MC.

Computationally, MC is quadratic in the number of models as it requires all pairwise comparisons. We also experiment with a lightweight version, called MC_S, where we randomly sample $P \leq N$ models and compute MC_i of M_i as the average of its similarities to P models, effectively reducing its complexity down to that of UDR.

In their experiments, [25] report that MC outperforms UDR schemes (Sec. *3.2.1*). Our results are consistent with their finding, possibly because it is an unrealistic hypothesis for outlier models that a good model would have consistent results across HP settings.

3.2.3 Model Centrality by HITS

We can build on the idea of ModelCentrality through computing centrality in a network setting. Unlike MC that is computed in one shot, network centrality is *recursive*—wherein a node has higher centrality the more they point to nodes that are pointed by other high-centrality ones.

One of the earliest methods for computing centrality, namely hubness h_p and authority a_p, of pages on the Web is the HITS algorithm [16], where

$$h_p \propto \text{sum of } a_i \text{ for all nodes } i \text{ that } p \text{ points to , and}$$

$$a_p \propto \text{sum of } h_i \text{ for all nodes } i \text{ pointing to } p \text{ ,}$$

which are estimated alternatingly over iterations until convergence. Besides ranking on the Web, HITS-like ideas have been used to estimate user trustworthiness in online rating platforms [20; 44], physician authoritativeness in patient referral networks [32], polarity of subjects in political networks [3], as well as truth discovery [47].

It is easy to adapt HITS for UOMS by constructing a complete bipartite network between the N models and n_t samples in a given dataset D_t. Then, the models can be evaluated by their hubness centralities. The analogous interpretation is that a a sample has higher authority (outlierness), the more trusted models (with high hubness) point to it (with large outlierness score, i.e. large edge weight). Then, a model is more central or trusted, the more it points (with large outlierness score) to samples with high authority.

Note that a by-product of this strategy is a consensus-based ranking of the samples based on authority scores (i.e. centrality-based outlierness) upon convergence. We compare this (aggregate) ranking, called HITS-AUTH, against selecting a (single) model by hubness in the experiments.

3.2.4 Unsupervised outlier model ensembling

HITS has a built-in advantage that is the iterative refinement of model trustworthiness. Specifically, given the trustworthiness of models, outlier scores can be better estimated by a trustworthiness-weighted aggregation of scores across models. Then, given those refined outlier scores, model trustworthiness can also be better estimated; where the more similar their output is to the updated scores, the more a model is deemed trustworthy.

Here we build on another iterative scheme, originally designed for unsupervised selective outlier model ensembling [38; 52]. The idea is to infer reliable "pseudo ground truth" outlier scores via aggregating the output of a carefully-selected subset of trustworthy models. The ensemble is constructed bottom-up in a greedy iterative fashion (see Alg. 1).

Algorithm 1 Ensemble-based Internal Model Evaluation

Input: set of outlier scores from all models, $\{\mathbf{s}_i\}_{i=1}^N$
Output: internal scores for all models

$\mathcal{S} := \emptyset$, $\mathcal{E} := \emptyset$, $C := 0$
for $i = 1, \ldots, N$ **do** ▶ convert scores to inverse rank
  $\mathcal{S} := \mathcal{S} \cup \{1/\text{rank}(s_{ij})\}_{j=1}^n$
end for
$target := \text{avg}(\mathcal{S})$ ▶ initial pseudo ground truth scores
repeat
  sort $\mathcal{S}$ by rank *corr*elation to $target$ in desc. order
  $\{m, \text{corr}_m\} := \text{fetchFirst}(\mathcal{S})$
  if $corr(\text{avg}(\mathcal{E} \cup m), target) \times |E| \geq C$ **then**
    $\mathcal{E} := \mathcal{E} \cup m, \quad C \mathrel{+}= \text{corr}_m$
    $target := \text{avg}(\mathcal{E})$ ▶ pseudo ground truth by $\mathcal{E}$
  end if
until $\{\mathcal{S} = \emptyset$ or $\mathcal{E}$ is not updated$\}$
return rank *corr*elation of $\mathbf{s}_i$ to $target$, $i = 1, \ldots, N$

Similar to HITS, the "pseudo ground truth" and model trustworthiness are estimated alternatingly. The latter is computed as the ranking based similarity of a model's output to the "pseudo ground truth" (i.e. $target$ in Alg. 1) at a given iteration. We adapt this framework to UOMS by using these similarities at convergence to evaluate the models. We call this strategy ENS. In experiments, we also compare the (aggregate) ranking by the ensemble (based on $target$), called ENS-PSEUDO, to selecting a (single) model (with highest similarity to $target$).

To wrap up, we give a summary of the 7 families of UOMS techniques as described in this section in Table 2.

4. RELATED WORK

In the outlier mining literature, several evaluation and benchmarking surveys draw attention to the fact that most (classical) outlier detectors are sensitive to their HPs [1; 6; 13; 15]. This is even more so for the recently booming deep learning based detection methods, as we empirically studied recently [9]. Despite its critical importance, however, related work on unsupervised outlier model selection (UOMS) is slim, with only a few existing works that address the problem.

In this survey, we focus on internal model evaluation strategies, which we reviewed in the previous section. In the following, we provide a critique and comparison between them. More recently other novel approaches have been proposed for UOMS, leveraging two main themes, meta-learning and ensemble methods, which we also review for completeness.

Internal model evaluation strategies for UOMS: Cluster quality based measures [34] and statistical mass based EM/MV methods [11] rely only on output scores. In contrast IREOS [29; 30] uses more information, that is both outlier scores and the original input samples (See Eq. (1)). Verifying that outlier scores align (correlate) with the separability of samples in the feature space is potentially less error-prone than simply looking at whether outlier/inlier scores are well clustered or separated – e.g., a model that outputs a $\{0, 1\}$ score per point at random would be considered a good model by the latter. The trade-off is the computational overhead for quantifying separability per sample.

In their work, IREOS is employed for UOMS using only 2 detectors (LOF [4] and kNN [36]), each with 17 different HP configurations (for a total of 34 models) on 11 datasets.

Being the seminal work, there is no comparison to any other techniques (existing or adapted). [34] acknowledge IREOS and criticize its computational demand, without any comparison. They also do not perform any UOMS in experiments, rather, they study the decay in internal measures as the ground truth ranking is contaminated via random swaps at the top based on 12 datasets. Finally, [11] performs UOMS using only and exactly 3 models (LOF, iForest [27], OCSVM [41]), each with a single (unspecified) HP configuration, on 8 datasets. None of these three compares to any other in their work. Moreover, because the datasets, experimental design, and the model pool specified by each work is different, it is not possible to do any direct comparison. In this work, we do a systematic comparison for the first time, using a much larger testbed (8 detectors, 297 models, 39 datasets) than originally considered by any prior work.

Other internal model evaluation strategies repurposed for UOMS: All three existing methods for UOMS are stand-alone, evaluating a model independently from the others. Having trained all models among which to select from, it is reasonable to take advantage of the similarities/agreement among them. To this end, we have repurposed methods from unsupervised representation learning [10; 25], network centrality [16], and unsupervised ensemble learning [38; 52] all of which are based on the "collective intelligence" of the models in the pool.

As we will show in the experiments, these strategies produce superior outcomes than existing, stand-alone methods. As such, our study motivates future work on consensus-based strategies, and calls for the transfer of prominent ideas from other similar fields, such as truth discovery and crowdsourcing, toward tackling the important problem of UOMS.

Recent novel approaches to UOMS: Most recently, two promising new directions have been explored toward UOMS. The first idea is building *hyper*-ensembles [9], which combines the outlier scores from multiple models with various HP configurations, rather than trying to select a single one of them. They have shown that the hyper-ensemble is significantly more robust to its own HPs (namely, the number of models to assemble and the value range per HP). The key challenge is similar to internal strategies covered in this paper, specifically, training all the models for assembly at test time is expensive, for which several speed up techniques have been proposed in [9].

The second line of work leverages *meta-learning* [48; 49; 51], where a database of historical outlier detection tasks *with labels* are used to transfer "knowledge/experience" toward UOMS on similar test tasks (without labels). Interestingly, internal evaluation measures have been exploited in (meta-)learning a mapping from such weak internal signals, dataset characteristics, etc. onto model performance (which can be computed for labeled historical tasks). Such a mapping is then employed for model performance prediction on test tasks without the need to access labels. This suggests that new internal evaluation measures or any improvements that lead to stronger internal signals of performance are to boost these meta-learning based solutions to UOMS. Computationally, meta-learning approaches are also more feasible than hyper-ensembles, as most computation is off-loaded to the meta-learning phase while fewer models are trained at test time. Of course, fast yet effective internal measures would contribute to further speed up model selection on a new test task.

5. EMPIRICAL EVALUATION OF INTERNAL MODEL EVALUATION STRATEGIES

5.1 Setup

Datasets and Model Pool. We already discussed the real-world datasets and candidate models of this study in Sec. 2. We build the experiments on **39 widely used outlier detection benchmark dataset**. As shown in Table 3, 21 datasets are from the **ODDS Library** [37], and the other 18 datasets are from **DAMI datasets** [6]. The specifications for all N=297 models have been listed in Sec. 2 Table 1.

Table 3: Real-world dataset pool composed by ODDS library (21 datasets) and DAMI library (18 datasets).

	Dataset	Num Pts	Dim	% Outlier
1	annthyroid (ODDS)	7200	6	7.416
2	arrhythmia (ODDS)	452	274	14.601
3	breastw (ODDS)	683	9	34.992
4	glass (ODDS)	214	9	4.205
5	ionosphere (ODDS)	351	33	35.897
6	letter (ODDS)	1600	32	6.250
7	lympho (ODDS)	148	18	4.054
8	mammography (ODDS)	11183	6	2.325
9	mnist (ODDS)	7603	100	9.206
10	musk (ODDS)	3062	166	3.167
11	optdigits (ODDS)	5216	64	2.875
12	pendigits (ODDS)	6870	16	2.270
13	pima (ODDS)	768	8	34.895
14	satellite (ODDS)	6435	36	31.639
15	satimage-2 (ODDS)	5803	36	1.223
16	speech (ODDS)	3686	400	1.654
17	thyroid (ODDS)	3772	6	2.465
18	vertebral (ODDS)	240	6	12.500
19	vowels (ODDS)	1456	12	3.434
20	wbc (ODDS)	378	30	5.555
21	wine (ODDS)	129	13	7.751
22	Annthyroid (DAMI)	7129	21	7.490
23	Arrhythmia (DAMI)	450	259	45.777
24	Cardiotocography (DAMI)	2114	21	22.043
25	HeartDisease (DAMI)	270	13	44.444
26	InternetAds (DAMI)	1966	1555	18.718
27	PageBlocks (DAMI)	5393	10	9.456
28	Pima (DAMI)	768	8	34.895
29	SpamBase (DAMI)	4207	57	39.909
30	Stamps (DAMI)	340	9	9.117
31	Wilt (DAMI)	4819	5	5.333
32	ALOI (DAMI)	49534	27	3.044
33	Glass (DAMI)	214	7	4.205
34	PenDigits (DAMI)	9868	16	0.202
35	Shuttle (DAMI)	1013	9	1.283
36	Waveform (DAMI)	3443	21	2.904
37	WBC (DAMI)	223	9	4.484
38	WDBC (DAMI)	367	30	2.724
39	WPBC (DAMI)	198	33	23.737

Baselines. We compare the model selected by each technique (Sec. 3) to two baselines across datasets.

- RANDOM, whose performance is the average of all (297) models per dataset. This is equivalent to expected performance when selecting a model from the candidate pool at random.
- IFOREST-R, with performance as the average of all (81) iForest models in the pool, equivalent to using iForest [27] (a state-of-the-art ensemble detector) with randomly chosen hyperparameters.[8]

[8]Family-wise performances across datasets in Supp. A show that iForest is the most competitive among the 8 families of detectors in this study, and thus the strongest baseline.

Method Configurations. For clustering-quality based measures, we split into two clusters as the top o_t (true number of outliers) and the rest, i.e. give those strategies the advantage of knowing o_t. This avoids the clustering step, which requires us to pick a clustering algorithm etc. and directly focuses on the measures themselves.

For EM and MV[9], we use the default values for α and t respectively (See Sec. *3.1.2*) and set `n_generated`$= 100K$, which is the number of random samples to generate for estimating the null distributions.

For IREOS, we use the author recommended settings;[10] $\gamma_{\max}$:=findGammaMaxbyDistances($\cdot$) with `sampling`=100, `tol`$=5 \times 10^{-3}$, and `clump_size`=10.

For UDR, MC, and MC$_S$, we experiment with three different pairwise similarity measures: Spearman's ρ, Kendall's τ, and NDCG [26]. For MC$_S$, $P = \sqrt{N} \approx 18$.

For HITS and ENS, we set edge weights between model M_i and sample j in a dataset as $1/r_{ij}$, where r_{ij} is the position of j in the rankedlist by M_i. Raw outlier scores are not used as they are not comparable across models. For comparison between selection versus consensus/ensembling, we also report the performance of the consensus outcome, called HITS-AUTH and ENS-PSEUDO; as ranked (resp.) by authority scores and by the pseudo ground truth at convergence.

Performance metrics. We evaluate performance w.r.t. three metrics. Two are based on the ranking quality: Average Precision (**AP**): the area under the precision-recall curve and **ROC** AUC: the area under the recall-false positive rate curve. The third metric measures the quality at the top: **Prec@*k***, precision at top k where we set $k = o_t$ (i.e. true number of outliers) for each $D_t \in \mathcal{D}$. In Supp. B we show that performances vary considerably across models for most datasets, justifying the importance of model selection.

For brevity, all results in this section are w.r.t. AP. Corresponding results for other metrics are similar, all of which are provided in Supp. C.

5.2 Results

5.2.1 Cluster quality based methods

We start by studying the 10 cluster quality based methods to identify those that stand out. We report the p-values by the one-sided[11] paired Wilcoxon signed rank test in Table 4. STD is significantly worse than all other methods. Three strategies that stand out are RS, CH, and XB, which are identical; in the sense that despite differences in their values and overall ranking, they select exactly the same model on each dataset. Importantly, while both STD and S are significantly worse than RANDOM at $p = 0.05$, none of the others is significantly different from RANDOM (!) All methods (including XB, RS, and CH) are significantly worse than IFOREST-R.

These findings suggest that cluster quality based internal evaluation methods would not be useful for UOMS.

5.2.2 Other stand-alone methods

As discussed in Sec. *3.1.2*, EM and MV quantify (roughly) the clusteredness of the inlier scores. Therefore, they are conceptually similar to the clustering quality based methods. Our findings confirm this intuition. As shown in Table 5, there is no significant difference between EM/MV and XB/RS/CH or RANDOM. Both of them are also significantly worse than IFOREST-R. Thus, they do not prove useful for UOMS. Findings are similar for IREOS; despite using more information (input samples besides scores, see Eq. (1)) and computational cost, it is only comparable to RANDOM.

Table 4: Comparison of cluster quality based methods and baselines by one-sided paired Wilcoxon signed rank test. p-values **bolded** (underlined) highlight the cases where the "row-method" is significantly **better** (worse) than the "column-method" at $p \leq 0.05$.

	STD	H	S	I	DB	SD	D	RND	IF
XB,RS,CH	**0.004**	0.240	**0.038**	0.212	0.370	0.127	0.357	0.500	0.981
STD		0.997	0.961	0.997	0.982	0.967	0.999	1.000	1.000
H			0.373	0.500	0.725	0.379	0.675	0.849	0.996
S				0.627	0.949	0.557	0.881	0.953	0.999
I					0.730	0.384	0.742	0.882	0.997
DB						0.307	0.647	0.522	0.982
SD							0.823	0.910	0.995
D								0.572	0.990
RND									1.000

Table 5: Comparison of stand-alone methods and baselines.

	EM	MV	IREOS	RND	IF
XB,RS,CH	0.533	0.500	0.862	0.500	0.981
EM		0.079	0.642	0.539	0.979
MV			0.716	0.687	0.994
IREOS				0.303	0.908

We provide an additional viewpoint by identifying the *q-th best* model per dataset where there exists no significant difference between the performance of the q-th best model and that selected by a given UOMS strategy across datasets. We report the *smallest* q for which one-sided Wilcoxon signed rank test yields p>0.05 in Table 6. A method with smaller q is better; the interpretation being that it could select, from a pool of 297, the model that is as good as the q-th best model per dataset. Stand-alone methods do not fare well against IFOREST-R which is comparable to the 84-th best model.

5.2.3 Consensus-based methods

We first study one-shot methods UDR, MC, and MC$_S$ based on different similarity measures. As shown in Table 6, all versions provide similar results, which are significantly better than RANDOM, and not different from IFOREST-R. We note that the faster, sampling-based MC$_S$ achieves similar performance to MC and can be used as a practical alternative.

Iterative methods HITS and ENS produce similar results to these simple one-shot methods, despite aiming to refine estimates of model trustworthiness over iterations. Again, as shown in Table 6, they significantly outperform RANDOM and are comparable to IFOREST-R. The same holds true for their respective consensus scores, HITS-AUTH and ENS-PSEUDO, where model aggregation provides no significant advantage over selecting the best (single) model.

Table 7 shows a pairwise comparison of the consensus-based methods by one-sided Wilcoxon signed rank test, confirming mostly no significant difference between them.

5.2.4 Running time analysis

In Fig. 1 we present for each method the running times on all

[9]Code available at `https://github.com/ngoix/EMMV_benchmarks`

[10]We thank Henrique Marques who helped with running their source code, `https://github.com/homarques/ireos-extension`

[11]Testing the hypothesis: row-method is better than col-method (against the null hypothesis stating no difference). For reverse order, p-value = 1 minus the reported value.

Table 6: **Summary of results**: p-values by one-sided paired Wilcoxon signed rank test comparing UOMS methods to the baselines, smallest q-th best model with no significant difference, and mean/standard deviation AP across datasets.

	Method	RANDOM	IFOREST-R	q_{AP}	mean AP	std AP
S-alone	XB,RS,CH	0.500	0.981	127	0.354	0.298
	EM	0.539	0.979	115	0.322	0.265
	IREOS	0.303	0.908	99	0.335	0.261
Consensus-based	UDR-ρ	**0.012**	0.905	104	0.383	0.283
	UDR-τ	**0.019**	0.952	109	0.379	0.282
	UDR-$NDCG$	**0.004**	0.825	93	0.384	0.270
	MC-ρ	**0.000**	0.217	89	0.395	0.289
	MC-τ	**0.002**	0.062	81	0.396	0.297
	MC-$NDCG$	**0.000**	0.182	82	0.404	0.291
	MC$_S$-ρ	**0.007**	0.706	108	0.385	0.289
	MC$_S$-τ	**0.001**	0.599	90	0.397	0.305
	MC$_S$-$NDCG$	**0.001**	0.205	83	0.391	0.285
	HITS	**0.000**	0.494	95	0.397	0.299
	ENS	**0.002**	0.730	81	0.371	0.282
Agg.	HITS-AUTH	**0.000**	0.577	94	0.401	0.286
	ENS-PSEUDO	**0.001**	0.422	79	0.373	0.282
Base.	RANDOM	–	1.000	144	0.342	0.234
	IFOREST-R	–	–	84	0.399	0.300

Table 7: Comparison of consensus-based methods (UDR, MC, MC$_S$ are based on $NDCG$).

	MC	MC$_S$	HITS	ENS
UDR	0.810	0.364	0.739	0.400
MC		0.551	**0.039**	0.116
MC$_S$			0.296	0.369
HITS				0.753

datasets.[12] IREOS and EM/MV are both computationally demanding, while ineffective. In fact, IREOS takes more than 16 days (!) on the largest dataset (ALOI), due to kernel SVM training *for each* sample. MC is the next most expensive method, which is quadratic in the number of models, although still takes less than 1 hr on ALOI. In short, MC$_S$, ENS, and especially HITS prove to be both competitive as well as fast UOMS methods, completing within 10 minutes on our testbed.

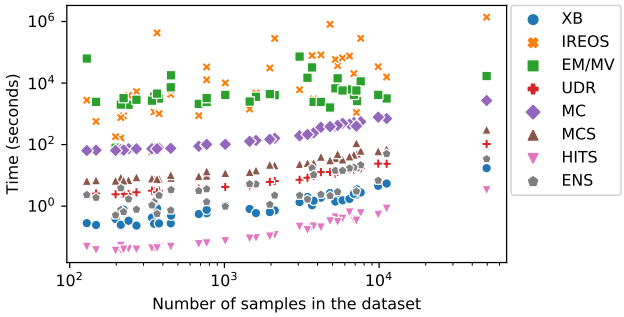

Figure 1: Run time comparison of UOMS methods.

5.3 Discussion of the Results

We conclude with the two key take-aways from our study:

1. *None of the existing (stand-alone) UOMS methods is significantly different from random model selection* (!), and with the exception of IREOS. All are significantly worse than iForest (with random hyperparameter configuration). The slight advantage of IREOS can be attributed to it utilizing input features in addition to model outlier scores, unlike the other strategies that solely use the output scores. However, this advantage comes at the expense of significant running time.

2. *All consensus-based methods that we repurposed for UOMS are significantly better than random selection, but not different from the fast, state-of-the-art iForest detector with default HPs.*

[12] On an Intel Xeon E7 4830 v3 @ 2.1Ghz with 1TB RAM.

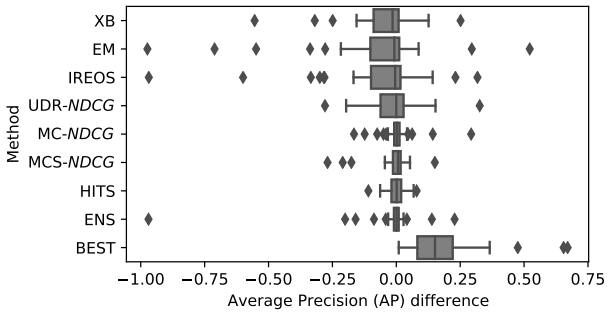

Figure 2: Distribution of performance difference across datasets: AP of selected model (by each UOMS method studied) minus that of IFOREST-R. Stand-alone methods and UDR are subpar, whereas other consensus-based method differences concentrate around zero (indicating no notable difference from IFOREST-R). Also shown for comparison is BEST model on each dataset, showcasing ample room for improvement over IFOREST-R.

Fig. 2 illustrates these take-aways where we show, via boxplots, the distribution of the performance difference between the model selected by each UOMS method and IFOREST-R across datasets. Consensus-based methods select models at best as good as IFOREST-R, where the AP difference concentrates around zero, whereas others are inferior.

These results suggest that **none of the UOMS methods we studied would be useful in practice**; because one would not first train a large *pool* of models – which would incur considerable computation – and then run a post hoc UOMS method to select a model, only to achieve comparable performance to a *single* iForest model (with default configuration) – which, in contrast, is extremely fast to train as it builds randomized trees on subsamples of data.

However, this is not to conclude that iForest is the best that one can hope to do. As given in Table 6, IFOREST-R is only as good as the 84-th best model per dataset. While it is the most competitive detector on average, other families outperform iForest on 28 out of 39 datasets in our study w.r.t. AP (See Table 16 in Supp. A, also see Tables 17 and 18 respectively for ROC and Prec@k). In Fig. 2 we also show the performance difference of the 1-st BEST model per dataset from IFOREST-R. (Also see Fig 3 in Supp. C.) One can clearly recognize that there is considerable room for progress in the area of UOMS.

6. CONCLUSION & CALL FOR FUTURE WORK ON UOMS

In this review, we considered the unsupervised outlier model selection (UOMS) problem: Given an unlabeled outlier de-

tection task, which detection algorithm and associated hyperparameter (HP) settings should one use? This is a question of utmost importance not only for practitioners to do well on their new task, but also for the research community for being able to fairly compare new detection methods and keep an accurate track record of progress in the field. On the other hand, the problem is notoriously hard in the absence of any labeled data, any well-accepted objective or loss function, and potentially very large model space especially for deep outlier detectors with many HPs.

We focused on the body of methods that proposed internal (i.e., unsupervised) model evaluation strategies that leverage implicit signals from the input features and /or the output outlier scores alone. On a large testbed comprising 297 models and 39 real-world datasets, we evaluated 7 different families of such internal evaluation strategies against simple baselines. Strikingly, we found that while consensus-based strategies are more promising against stand-alone ones which are not significantly better than random, none of them provides significant improvement over the state-of-the-art iForest detector with default HPs.

Our findings call for further research in this important area. As our work recently showed [9], deep detectors are considerably poor across varying HPs *on average* (i.e. when HPs are chosen randomly in the absence of any other guidance). As such, UOMS appears to stand as the biggest obstacle in front of deep models to fulfill their potential for outlier detection. A promising future direction is to develop stronger and faster internal strategies that can be leveraged within a meta-learning framework as in [48; 49]. Our empirical evaluation revealed consensus-based internal strategies to be relatively more promising, which provides fertile ground for adaptation of prominent ideas from related areas such as truth discovery and crowdsourcing. To foster progress on this critical problem, we publicly share all source code, trained models, and datasets at `https://github.com/yzhao062/uoms`.

7. REFERENCES

[1] C. C. Aggarwal and S. Sathe. Theoretical foundations and algorithms for outlier ensembles. *Acm sigkdd explorations newsletter*, 17(1):24–47, 2015.

[2] C. C. Aggarwal and S. Sathe. *Outlier Ensembles: An Introduction*. Springer Publishing Company, Inc., 1st edition, 2017.

[3] L. Akoglu. Quantifying political polarity based on bipartite opinion networks. In *ICWSM*. The AAAI Press, 2014.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *SIGMOD*, pages 93–104, 2000.

[5] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *SIGMOD*, pages 93–104. ACM, 2000. SIGMOD Record 29(2), June 2000.

[6] G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *DMKD*, 30(4):891–927, July 2016.

[7] J. Chen, S. Sathe, C. C. Aggarwal, and D. S. Turaga. Outlier detection with autoencoder ensembles. In *SDM*, pages 90–98. SIAM, 2017.

[8] S. Das, W.-K. Wong, T. G. Dietterich, A. Fern, and A. Emmott. Incorporating expert feedback into active anomaly discovery. In *ICDM*, pages 853–858. IEEE Computer Society, 2016.

[9] X. Ding, L. Zhao, and L. Akoglu. Hyperparameter sensitivity in deep outlier detection: Analysis and a scalable hyper-ensemble solution. In *Advances in Neural Information Processing Systems*, 2022.

[10] S. Duan, L. Matthey, A. Saraiva, N. Watters, C. Burgess, A. Lerchner, and I. Higgins. Unsupervised model selection for variational disentangled representation learning. In *ICLR*. OpenReview.net, 2020.

[11] N. Goix. How to evaluate the quality of unsupervised anomaly detection algorithms? *CoRR*, abs/1607.01152, 2016.

[12] M. Goldstein and A. Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, pages 59–63, 2012.

[13] M. Goldstein and S. Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[14] S. Guha, N. Mishra, G. Roy, and O. Schrijvers. Robust random cut forest based anomaly detection on streams. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2712–2721, New York, New York, USA, 20–22 Jun 2016. PMLR.

[15] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao. ADBench: Anomaly detection benchmark. In *Advances in Neural Information Processing Systems*, 2022.

[16] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, Sept. 1999.

[17] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. *VLDB J.*, 8(3-4):237–253, 2000.

[18] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Interpreting and unifying outlier scores. In *SDM*, pages 13–24. SIAM / Omnipress, 2011.

[19] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *SIGKDD*, pages 444–452, 2008.

[20] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. S. Subrahmanian. Rev2: Fraudulent user prediction in rating platforms. In *WSDM*, pages 333–341. ACM, 2018.

[21] H. Lamba and L. Akoglu. Learning on-the-job to re-rank anomalies from top-1 feedback. In *SDM*, pages 612–620. SIAM, 2019.

[22] Y. Li, Z. Chen, D. Zha, K. Zhou, H. Jin, H. Chen, and X. Hu. Autood: automated outlier detection via curiosity-guided search and self-imitation learning. *arXiv preprint arXiv:2006.11321*, 2020.

[23] Y. Li, D. Zha, P. Venugopal, N. Zou, and X. Hu. Pyodds: An end-to-end outlier detection system with automated machine learning. In *Companion Proceedings of the Web Conference 2020*, pages 153–157, 2020.

[24] J. Liang and S. Parthasarathy. Robust contextual outlier detection: Where context meets sparsity. In *CIKM*, pages 2167–2172. ACM, 2016.

[25] Z. Lin, K. Thekumparampil, G. Fanti, and S. Oh. InfoGAN-CR and ModelCentrality: Self-supervised model training and selection for disentangling GANs. In *International Conference on Machine Learning*, pages 6127–6139. PMLR, 2020.

[26] C. Lioma, J. G. Simonsen, and B. Larsen. Evaluation measures for relevance and credibility in ranked lists. In *ICTIR*, pages 91–98. ACM, 2017.

[27] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *ICDM*, pages 413–422. IEEE, 2008.

[28] E. A. Manzoor, H. Lamba, and L. Akoglu. xstream: Outlier detection in feature-evolving data streams. In *KDD*, pages 1963–1972. ACM, 2018.

[29] H. O. Marques, R. J. G. B. Campello, J. Sander, and A. Zimek. Internal evaluation of unsupervised outlier detection. *ACM Trans. Knowl. Discov. Data*, 14(4):47:1–47:42, 2020.

[30] H. O. Marques, R. J. G. B. Campello, A. Zimek, and J. Sander. On the internal evaluation of unsupervised outlier detection. In *SSDBM*, pages 7:1–7:12. ACM, 2015.

[31] M. M. Meghanath, D. Pai, and L. Akoglu. Conout: Contextual outlier detection with multiple contexts: Application to ad fraud. In *ECML/PKDD (1)*, volume 11051, pages 139–156. Springer, 2018.

[32] A. Mishra, J. S. Pudipeddi, and L. Akoglu. Ranking in heterogeneous networks with geo-location information. In *SDM*, pages 408–416. SIAM, 2017.

[33] T. T. Nguyen, A. T. Nguyen, T. A. H. Nguyen, L. T. Vu, Q. U. Nguyen, and L. D. Hai. Unsupervised anomaly detection in online game. In *Proceedings of the Sixth International Symposium on Information and Communication Technology*, pages 4–10. ACM, 2015.

[34] V. Nguyen, T. Nguyen, and U. Nguyen. An evaluation method for unsupervised anomaly detection algorithms. *Journal of Computer Science and Cybernetics*, 32(3):259–272, 2017.

[35] T. Pevnỳ. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304, 2016.

[36] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *SIGMOD*, pages 427–438, 2000.

[37] S. Rayana. ODDS library, 2016.

[38] S. Rayana and L. Akoglu. Less is more: Building selective anomaly ensembles. *ACM Trans. Knowl. Discov. Data*, 10(4):42:1–42:33, 2016.

[39] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, Nov. 1987.

[40] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller. A unifying review of deep and shallow anomaly detection. *CoRR*, abs/2009.11732, 2020.

[41] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[42] S. C. Tan, K. M. Ting, and F. T. Liu. Fast anomaly detection for streaming data. In *IJCAI*, pages 1511–1516. IJCAI/AAAI, 2011.

[43] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *PAKDD*, pages 535–548. Springer, 2002.

[44] G. Wang, S. Xie, B. Liu, and P. S. Yu. Review graph based online store review spammer detection. In *ICDM*, pages 1242–1247. IEEE Computer Society, 2011.

[45] R. Wang, K. Nie, T. Wang, Y. Yang, and B. Long. Deep learning for anomaly detection. In *WSDM*, pages 894–896. ACM, 2020.

[46] X. L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(8):841–847, 1991.

[47] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. In *KDD*, pages 1048–1052. ACM, 2007.

[48] Y. Zhao and L. Akoglu. Toward unsupervised outlier model selection. *IEEE ICDM*, 2022.

[49] Y. Zhao and L. Akoglu. Towards unsupervised hyperparameter optimization for outlier detection. *arXiv preprint arXiv:2208.11727*, 2022.

[50] Y. Zhao, Z. Nasrullah, and Z. Li. Pyod: A python toolbox for scalable outlier detection. *JMLR*, 20(96):1–7, 2019.

[51] Y. Zhao, R. Rossi, and L. Akoglu. Automatic unsupervised outlier model selection. *Advances in Neural Information Processing Systems*, 34:4489–4502, 2021.

[52] A. Zimek, R. J. G. B. Campello, and J. Sander. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *SIGKDD Explor.*, 15(1):11–22, 2013.

APPENDIX

A. FAMILY-WISE MODEL PERFORMANCES

In this study we use **8 different families of outlier detection algorithms**, namely; LODA, ABOD, iForest, kNN, LOF, HBOS, OCSVM, and COF. We build a total of **297 detection models** based on various hyperparameter (HP) configurations of these algorithms, as listed in Table 1.

Tables 16, 17, and 18 (resp. for AP, ROC AUC, and Prec@k) show the family-wise average performance of each detection algorithm (averaged over within-family models with different HP settings) on each dataset, as well as mean and standard deviation across datasets.

These show **iForest to be the most competitive detector**, which we compare to as a baseline to study *whether unsupervised model selection outperforms always using the same (state-of-the-art) detector.*

B. MODEL PERFORMANCES ON INDIVIDUAL DATASETS

Figures 4, 5, and 6 (resp. for AP, ROC AUC, and Prec@k) show the distribution of performances across all 297 models via boxplots for each dataset. For most datasets, there exists **considerable difference between the best and the worst performing model**—suggesting that effective model selection would be beneficial.

C. CORRESPONDING RESULTS BASED ON OTHER METRICS

For brevity, we reported all performance results in Evaluation (Sec. 5) based on Average Precision (**AP**). For completeness, we provide the results of the same analysis corresponding to **ROC** AUC and **Prec@k** metrics.

The conclusions are similar for these two metrics.

Cluster quality based methods. Specifically, Tables 8 and 12 present, resp. for ROC and Prec@k, the pairwise comparison of cluster quality based methods and the baselines (RANDOM and IFOREST-R). Three strategies RS, CH, and XB appear to stand out from others. However, none of the methods are not significantly different from (and few are sometimes worse than) RANDOM. Most of them are significantly worse than IFOREST-R, with otherwise a very large p-value.

Other stand-alone methods. Tables 9 and 13 present, resp. for ROC and Prec@k, the pairwise comparison of all the stand-alone methods (only RS, CH, and XB from above) and the baselines. We find that they are not different from each other or RANDOM—implying that **stand-alone model selection techniques would not be useful in practice**.

Consensus-based methods. Tables 10 and 14 show, resp. for ROC and Prec@k, that all consensus-based techniques, namely UDR, MC, MC$_S$, HITS, and ENS, are comparable to each other in terms of selection performance.

Finally, Tables 11 and 15 provide, resp. for ROC and Prec@k, a summary of the results for all the unsupervised model selection methods we studied. Main take-aways are: (1) **Consensus-based model selection methods are more competitive** than stand-alone methods, where all of them achieve **significantly better performance than** RANDOM **selection**. (2) Further, they are most often not different from IFOREST-R (a state-of-the-art detector) and sometimes even better (w.r.t. ROC). However, their absolute difference (i.e. effect size) is negligible as shown in Figure 3 for both ROC and Prec@k. Notably, their performance differences are not far from zero, suggesting that **consensus-based selection would also not be preferrable in practice**, since training a *single* IFOREST-R model is much faster over training a *pool* of models (with considerable running time overhead) to select from.

Table 8: Comparison of cluster quality based methods and baselines by one-sided paired Wilcoxon signed rank test on ROC AUC. p-values **bolded** (underlined) highlight the cases where row-method is significantly **better** (worse) than col-method at $p \leq 0.05$.

	STD	H	S	I	DB	SD	D	RND	IF
XB,RS,CH	**0.001**	0.407	**0.007**	0.389	0.272	0.099	0.518	0.358	0.980
STD		1.000	0.990	1.000	0.994	0.995	1.000	1.000	1.000
H			**0.021**	0.500	0.487	0.320	0.831	0.818	1.000
S				0.974	0.816	0.704	0.994	0.994	1.000
I					0.487	0.323	0.849	0.821	1.000
DB						0.368	0.815	0.662	0.996
SD							0.842	0.905	0.999
D								0.110	0.998
RND									1.000

Table 9: Comparison of stand-alone methods and baselines w.r.t. ROC AUC.

	EM	MV	IREOS	RND	IF
XB,RS,CH	0.364	0.422	0.934	0.358	0.980
EM		0.079	0.969	0.358	0.992
MV			0.977	0.369	0.997
IREOS				**0.006**	0.702

Table 10: Comparison of consensus-based methods (UDR, MC, MC$_S$ are based on $NDCG$) w.r.t. ROC AUC.

	MC	MC$_S$	HITS	ENS
UDR	0.462	0.070	0.408	0.232
MC		0.100	0.134	0.069
MC$_S$			0.681	0.511
HITS				0.740

Table 11: **Summary of results**: p-values by one-sided paired Wilcoxon signed rank test comparing UOMS methods to the baselines, smallest q-th best model with no significant difference, and mean/standard deviation ROC AUC across datasets.

	Method	RANDOM	IFOREST-R	$q_{\mathbf{ROC}}$	mean ROC	std ROC
S-alone	XB,RS,CH	0.358	0.980	138	0.690	0.206
	EM	0.358	0.992	142	0.682	0.216
	IREOS	**0.006**	0.702	83	0.730	0.203
Consensus-based	UDR-ρ	**0.000**	0.279	82	0.763	0.180
	UDR-τ	**0.000**	0.186	75	0.769	0.180
	UDR-$NDCG$	**0.000**	0.175	75	0.769	0.183
	MC-ρ	**0.000**	**0.036**	92	0.767	0.168
	MC-τ	**0.000**	**0.011**	91	0.769	0.167
	MC-$NDCG$	**0.000**	**0.034**	86	0.771	0.170
	MC$_S$-ρ	**0.000**	0.483	100	0.763	0.173
	MC$_S$-τ	**0.000**	0.121	94	0.761	0.167
	MC$_S$-$NDCG$	**0.000**	0.274	94	0.766	0.165
	HITS	**0.000**	0.148	97	0.762	0.169
	ENS	**0.000**	0.230	86	0.749	0.183
Agg.	HITS-AUTH	**0.000**	**0.018**	77	0.785	0.163
	ENS-PSEUDO	**0.000**	0.135	87	0.749	0.184
Base.	RANDOM	–	1.000	183	0.704	0.133
	IFOREST-R	–	–	102	0.763	0.166

Table 12: Comparison of cluster quality based methods and baselines by one-sided paired Wilcoxon signed rank test on Prec@k. p-values **bolded** (underlined) highlight the cases where row-method is significantly **better** (worse) than col-method at $p \leq 0.05$.

	STD	H	S	I	DB	SD	D	RND	IF
XB,RS,CH	**0.000**	0.125	**0.031**	0.109	0.173	**0.026**	0.274	0.090	0.716
STD		0.998	0.985	0.999	0.989	0.956	1.000	1.000	1.000
H			0.447	0.704	0.623	0.191	0.581	0.500	0.967
S				0.488	0.815	0.313	0.875	0.757	0.961
I					0.631	0.203	0.632	0.544	0.978
DB						0.166	0.719	0.423	0.915
SD							0.929	0.879	0.993
D								0.201	0.923
RND									0.999

Table 13: Comparison of stand-alone methods and baselines w.r.t. Prec@k.

	EM	MV	IREOS	RND	IF
XB,RS,CH	0.272	0.193	0.405	0.090	0.716
EM		0.187	0.696	0.730	0.967
MV			0.770	0.829	0.987
IREOS				0.423	0.944

Table 14: Comparison of consensus-based methods (UDR, MC, MC$_S$ are based on $NDCG$) w.r.t. Prec@k.

	MC	MC$_S$	HITS	ENS
UDR	0.645	0.403	0.464	0.296
MC		0.145	0.227	0.341
MC$_S$			0.375	0.488
HITS				0.608

Table 15: **Summary of results**: p-values by one-sided paired Wilcoxon signed rank test comparing UOMS methods to the baselines, smallest q-th best model with no significant difference, and mean/standard deviation Prec@k across datasets.

	Method	RANDOM	IFOREST-R	q_{Prec}	mean Prec@k	std Prec@k
S-alone	XB,RS,CH	0.090	0.716	91	0.348	0.277
	EM	0.730	0.967	119	0.303	0.254
	IREOS	0.423	0.944	102	0.316	0.255
Consensus-based	UDR-ρ	**0.039**	0.965	115	0.354	0.271
	UDR-τ	**0.025**	0.942	110	0.356	0.263
	UDR-$NDCG$	**0.002**	0.600	86	0.372	0.255
	MC-ρ	**0.002**	0.555	98	0.369	0.271
	MC-τ	**0.002**	0.833	103	0.370	0.280
	MC-$NDCG$	**0.000**	0.228	89	0.378	0.270
	MC$_S$-ρ	**0.008**	0.937	115	0.361	0.276
	MC$_S$-τ	**0.002**	0.595	96	0.374	0.290
	MC$_S$-$NDCG$	**0.002**	0.210	92	0.367	0.274
	HITS	**0.001**	0.583	99	0.376	0.280
	ENS	**0.004**	0.595	92	0.351	0.261
Agg.	HITS-AUTH	**0.000**	0.293	89	0.380	0.263
	ENS-PSEUDO	**0.005**	0.722	89	0.350	0.262
Base.	RANDOM	–	0.999	153	0.325	0.217
	IFOREST-R	–	–	91	0.374	0.280

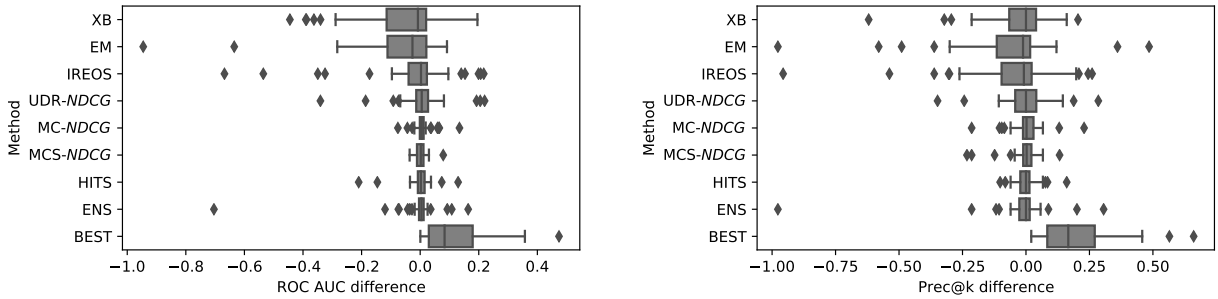

Figure 3: Distribution of performance difference across datasets: (left) ROC AUC and (right) Prec@k of selected model (by each UOMS method studied) minus that of iForest-r. Stand-alone methods and UDR are subpar, whereas consensus-based methods' differences concentrate around zero (not notably different from iForest-r). Also shown for comparison is Best model on each dataset, showcasing ample room for improvement over iForest-r.

Table 16: Family-wise model performance in AP. Values in **bold** highlight the model that outperforms for each dataset (per row). iForest achieves the highest average performance across all datasets.

Dataset	LODA	ABOD	iForest	kNN	LOF	HBOS	OCSVM	COF
annthyroid (ODDS)	0.136	0.232	0.340	0.228	0.172	**0.388**	0.145	0.138
arrhythmia (ODDS)	0.387	0.315	**0.470**	0.392	0.362	0.431	0.250	0.404
breastw (ODDS)	0.964	0.702	**0.972**	0.942	0.331	0.959	0.544	0.304
glass (ODDS)	0.063	0.137	0.104	0.106	0.117	0.061	0.063	**0.154**
ionosphere (ODDS)	0.766	**0.921**	0.784	0.868	0.819	0.288	0.492	0.852
letter (ODDS)	0.092	0.319	0.089	0.258	0.359	0.080	0.138	**0.459**
lympho (ODDS)	0.447	0.555	**0.957**	0.763	0.668	0.905	0.418	0.464
mammography (ODDS)	0.218	0.147	**0.234**	0.169	0.102	0.096	0.156	0.064
mnist (ODDS)	0.203	0.329	0.261	**0.401**	0.273	0.097	0.204	0.195
musk (ODDS)	0.904	0.038	0.990	0.588	0.130	**0.997**	0.498	0.174
optdigits (ODDS)	0.025	0.057	0.049	0.021	0.037	**0.177**	0.031	0.048
pendigits (ODDS)	0.245	0.057	**0.280**	0.104	0.038	0.231	0.086	0.037
pima (ODDS)	0.445	0.508	0.492	**0.524**	0.441	0.521	0.385	0.429
satellite (ODDS)	0.630	0.430	0.664	0.562	0.375	**0.711**	0.456	0.368
satimage-2 (ODDS)	0.904	0.212	**0.916**	0.615	0.055	0.717	0.486	0.078
speech (ODDS)	0.018	**0.093**	0.020	0.024	0.031	0.025	0.022	0.034
thyroid (ODDS)	0.238	0.218	0.587	0.354	0.157	**0.630**	0.196	0.032
vertebral (ODDS)	0.089	0.098	0.094	0.090	0.101	0.087	**0.131**	0.116
vowels (ODDS)	0.140	**0.690**	0.134	0.487	0.348	0.083	0.080	0.408
wbc (ODDS)	0.603	0.367	0.599	0.533	0.497	**0.673**	0.321	0.261
wine (ODDS)	0.286	0.082	0.215	0.253	0.253	**0.402**	0.249	0.081
Annthyroid (DAMI)	0.097	0.137	**0.160**	0.126	0.134	0.145	0.079	0.130
Arrhythmia (DAMI)	0.685	0.668	**0.757**	0.711	0.702	0.745	0.523	0.712
Cardiotocography (DAMI)	0.433	0.254	**0.433**	0.316	0.280	0.344	0.314	0.267
HeartDisease (DAMI)	0.562	0.547	0.538	0.557	0.509	**0.619**	0.475	0.486
InternetAds (DAMI)	0.251	0.293	0.490	0.289	0.263	**0.521**	0.237	0.261
PageBlocks (DAMI)	0.464	0.416	0.449	**0.526**	0.360	0.201	0.268	0.232
Pima (DAMI)	0.448	0.506	0.494	**0.529**	0.467	0.487	0.392	0.432
SpamBase (DAMI)	0.370	0.357	0.487	0.406	0.364	**0.532**	0.366	0.392
Stamps (DAMI)	0.332	0.218	**0.336**	0.313	0.228	0.315	0.209	0.159
Wilt (DAMI)	0.039	0.065	0.045	0.053	0.075	0.044	0.065	**0.101**
ALOI (DAMI)	0.034	0.102	0.033	0.057	0.100	0.031	0.035	**0.144**
Glass (DAMI)	0.085	**0.221**	0.183	0.146	0.118	0.115	0.107	0.179
PenDigits (DAMI)	0.003	0.031	0.005	**0.040**	0.014	0.004	0.016	0.017
Shuttle (DAMI)	0.111	0.250	0.071	**0.326**	0.296	0.094	0.095	0.173
Waveform (DAMI)	0.052	0.055	0.057	**0.115**	0.095	0.053	0.069	0.102
WBC (DAMI)	0.743	0.595	**0.858**	0.671	0.359	0.683	0.424	0.146
WDBC (DAMI)	0.720	0.296	0.669	0.571	0.554	**0.725**	0.322	0.295
WPBC (DAMI)	0.235	0.231	0.229	0.233	0.230	**0.239**	0.237	0.219
average	0.345	0.301	**0.399**	0.366	0.277	0.371	0.246	0.245
STD	0.282	0.220	0.304	0.248	0.199	0.295	0.165	0.188

Table 17: Family-wise model performance in ROC AUC. Values in **bold** highlight the model that outperforms for each dataset (per row). kNN (0.764) and iForest (0.763) achieve the highest average performance across all datasets.

Dataset	LODA	ABOD	iForest	kNN	LOF	HBOS	OCSVM	COF
annthyroid (ODDS)	0.572	0.823	**0.841**	0.775	0.729	**0.736**	0.517	0.689
arrhythmia (ODDS)	0.735	0.751	**0.803**	0.777	0.764	**0.806**	0.522	0.757
breastw (ODDS)	0.980	0.898	**0.988**	0.980	0.500	0.985	0.481	0.459
glass (ODDS)	0.539	0.766	0.707	0.747	0.747	0.638	0.429	**0.772**
ionosphere (ODDS)	0.814	**0.928**	0.838	0.898	0.870	0.357	0.548	0.879
letter (ODDS)	0.584	**0.880**	0.629	0.842	0.846	0.581	0.554	**0.880**
lympho (ODDS)	0.814	0.936	**0.998**	0.971	0.938	0.985	0.607	0.834
mammography (ODDS)	0.854	0.822	**0.862**	0.845	0.729	0.799	0.629	0.700
mnist (ODDS)	0.586	0.797	0.794	**0.856**	0.708	0.515	0.536	0.615
musk (ODDS)	0.991	0.072	0.999	0.830	0.521	**1.000**	0.669	0.534
optdigits (ODDS)	0.414	0.477	0.713	0.383	0.463	**0.877**	0.463	0.526
pendigits (ODDS)	0.934	0.692	**0.948**	0.818	0.516	0.921	0.548	0.508
pima (ODDS)	0.629	0.685	0.652	**0.717**	0.630	0.634	0.497	0.583
satellite (ODDS)	0.644	0.594	0.703	0.703	0.546	**0.785**	0.506	0.519
satimage-2 (ODDS)	0.988	0.854	**0.993**	0.965	0.678	0.973	0.610	0.537
speech (ODDS)	0.474	**0.688**	0.473	0.500	0.525	0.473	0.492	0.584
thyroid (ODDS)	0.820	0.945	**0.983**	0.960	0.771	**0.950**	0.550	0.581
vertebral (ODDS)	0.315	0.375	0.349	0.333	0.380	0.297	**0.482**	0.454
vowels (ODDS)	0.712	**0.976**	0.736	0.944	0.905	0.676	0.529	0.877
wbc (ODDS)	0.941	0.918	0.938	0.935	0.892	**0.950**	0.603	0.792
wine (ODDS)	0.853	0.490	0.794	0.779	0.758	**0.873**	0.536	0.373
Annthyroid (DAMI)	0.491	**0.717**	**0.679**	0.658	0.679	0.646	0.471	0.666
Arrhythmia (DAMI)	0.687	0.725	**0.750**	0.736	0.732	0.736	0.506	0.736
Cardiotocography (DAMI)	0.689	0.458	**0.689**	0.503	0.544	0.566	0.489	0.522
HeartDisease (DAMI)	0.608	0.612	0.602	0.637	0.582	**0.670**	0.502	0.542
InternetAds (DAMI)	0.548	0.657	0.690	0.626	0.587	**0.695**	0.499	0.579
PageBlocks (DAMI)	0.785	0.780	**0.894**	**0.889**	0.759	0.679	0.558	0.610
Pima (DAMI)	0.624	0.666	0.644	**0.706**	0.650	0.594	0.504	0.587
SpamBase (DAMI)	0.433	0.403	0.635	0.535	0.441	**0.676**	0.463	0.450
Stamps (DAMI)	0.891	0.793	**0.901**	0.872	0.702	0.876	0.582	0.541
Wilt (DAMI)	0.363	0.628	0.457	0.538	0.626	0.419	0.489	**0.695**
ALOI (DAMI)	0.504	0.739	0.534	0.641	0.744	0.508	0.506	**0.796**
Glass (DAMI)	0.659	**0.854**	0.794	0.822	0.748	0.795	0.485	0.774
PenDigits (DAMI)	0.628	0.936	0.768	**0.967**	0.821	0.734	0.537	0.718
Shuttle (DAMI)	0.637	0.927	0.853	**0.963**	0.911	0.842	0.566	0.848
Waveform (DAMI)	0.664	0.666	0.707	**0.743**	0.716	0.703	0.492	0.689
WBC (DAMI)	0.983	0.954	**0.991**	0.979	0.842	0.985	0.611	0.703
WDBC (DAMI)	0.945	0.890	0.936	0.924	0.871	**0.963**	0.629	0.800
WPBC (DAMI)	0.509	0.501	0.498	0.509	0.503	**0.536**	0.485	0.463
average	0.688	0.725	0.763	**0.764**	0.689	0.729	0.530	0.645
STD	0.188	0.197	0.168	0.175	0.146	0.188	0.054	0.138

Table 18: Family-wise model performance in Prec@k. Values in **bold** highlight the model that outperforms for each dataset (per row). iForest achieves the highest average performance across all datasets.

Dataset	LODA	ABOD	iForest	kNN	LOF	HBOS	OCSVM	COF
annthyroid (ODDS)	0.180	0.301	0.337	0.297	0.209	**0.387**	0.180	0.169
arrhythmia (ODDS)	0.403	0.372	0.481	0.411	0.386	**0.495**	0.237	0.407
breastw (ODDS)	0.924	0.788	0.929	0.923	0.271	**0.938**	0.445	0.152
glass (ODDS)	0.019	0.111	0.111	0.111	0.136	0.014	0.040	**0.143**
ionosphere (ODDS)	0.645	**0.849**	0.648	0.753	0.725	0.228	0.439	0.764
letter (ODDS)	0.100	0.354	0.092	0.312	0.358	0.080	0.140	**0.440**
lympho (ODDS)	0.401	0.476	**0.881**	0.639	0.560	0.808	0.347	0.405
mammography (ODDS)	**0.286**	0.197	0.261	0.251	0.194	0.114	0.192	0.114
mnist (ODDS)	0.212	0.376	0.293	**0.420**	0.315	0.095	0.218	0.246
musk (ODDS)	0.873	0.035	0.977	0.546	0.134	**0.981**	0.491	0.218
optdigits (ODDS)	0.001	0.045	0.025	0.000	0.029	**0.211**	0.018	0.067
pendigits (ODDS)	0.324	0.077	**0.365**	0.110	0.072	0.269	0.113	0.063
pima (ODDS)	0.466	0.530	0.504	**0.551**	0.463	0.476	0.361	0.423
satellite (ODDS)	0.533	0.417	0.573	0.511	0.379	**0.619**	0.382	0.361
satimage-2 (ODDS)	**0.865**	0.260	0.862	0.577	0.086	0.661	0.465	0.145
speech (ODDS)	0.019	**0.138**	0.031	0.039	0.045	0.032	0.039	0.049
thyroid (ODDS)	0.287	0.198	0.620	0.332	0.149	**0.645**	0.224	0.000
vertebral (ODDS)	0.011	0.043	0.044	0.018	0.056	0.012	0.074	**0.090**
vowels (ODDS)	0.194	**0.641**	0.175	0.474	0.333	0.121	0.094	0.429
wbc (ODDS)	0.558	0.361	0.536	0.496	0.475	**0.614**	0.324	0.293
wine (ODDS)	0.257	0.000	0.140	0.194	0.203	**0.408**	0.200	0.043
Annthyroid (DAMI)	0.116	0.153	**0.213**	0.134	0.165	0.191	0.074	0.162
Arrhythmia (DAMI)	0.604	0.630	**0.655**	0.637	0.643	0.632	0.459	0.652
Cardiotocography (DAMI)	**0.407**	0.266	0.396	0.311	0.288	0.303	0.259	0.264
HeartDisease (DAMI)	0.530	0.520	0.503	0.535	0.506	**0.591**	0.447	0.470
InternetAds (DAMI)	0.267	0.344	0.449	0.334	0.304	**0.466**	0.244	0.284
PageBlocks (DAMI)	0.458	0.425	0.397	**0.506**	0.376	0.158	0.264	0.268
Pima (DAMI)	0.476	0.512	0.499	**0.547**	0.485	0.448	0.369	0.421
SpamBase (DAMI)	0.351	0.359	0.518	0.421	0.338	**0.562**	0.357	0.382
Stamps (DAMI)	0.275	0.189	0.286	0.211	0.169	**0.385**	0.197	0.180
Wilt (DAMI)	0.001	0.012	0.012	0.003	0.058	0.006	0.043	**0.121**
ALOI (DAMI)	0.050	0.144	0.028	0.086	0.146	0.028	0.043	**0.187**
Glass (DAMI)	0.027	0.143	0.111	0.111	0.133	0.044	0.056	**0.159**
PenDigits (DAMI)	0.000	0.036	0.000	0.000	0.019	0.000	0.010	**0.036**
Shuttle (DAMI)	0.120	**0.319**	0.079	0.277	0.169	0.092	0.092	0.231
Waveform (DAMI)	0.057	0.069	0.065	**0.191**	0.161	0.063	0.083	0.143
WBC (DAMI)	0.630	0.429	**0.723**	0.644	0.328	0.713	0.356	0.086
WDBC (DAMI)	**0.650**	0.271	0.633	0.592	0.536	0.648	0.350	0.286
WPBC (DAMI)	0.166	0.164	0.146	0.160	0.172	**0.206**	0.202	0.161
average	0.327	0.296	**0.374**	0.350	0.271	0.352	0.229	0.244
STD	0.262	0.214	0.284	0.235	0.180	0.283	0.149	0.171

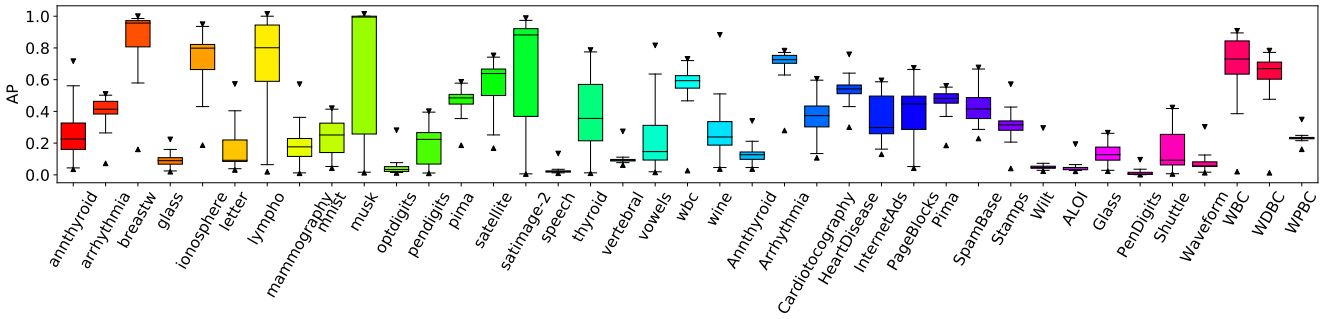

Figure 4: Model performance boxplot (AP) for all datasets, where triangles mark the min and max. Model performance varies significantly for most datasets, showing the importance of model selection.

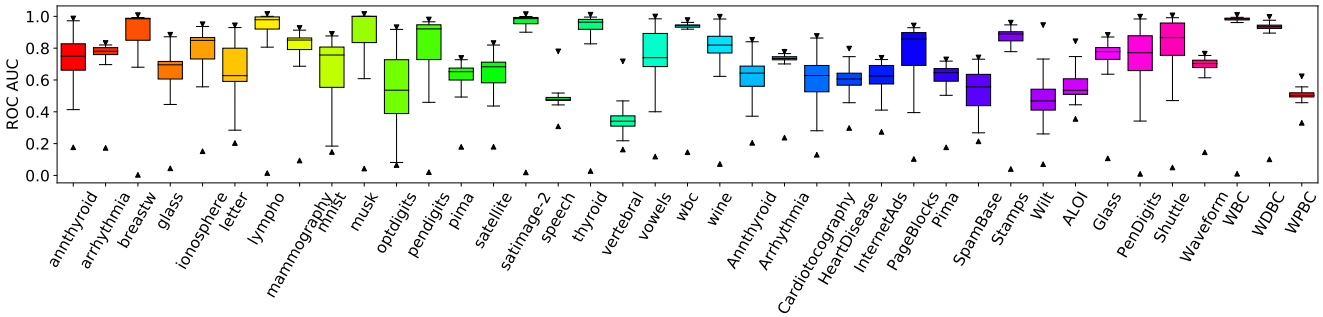

Figure 5: Model performance boxplot (ROC AUC) for all datasets, where triangles mark the min and max. Model performance varies significantly for most datasets, showing the importance of model selection.

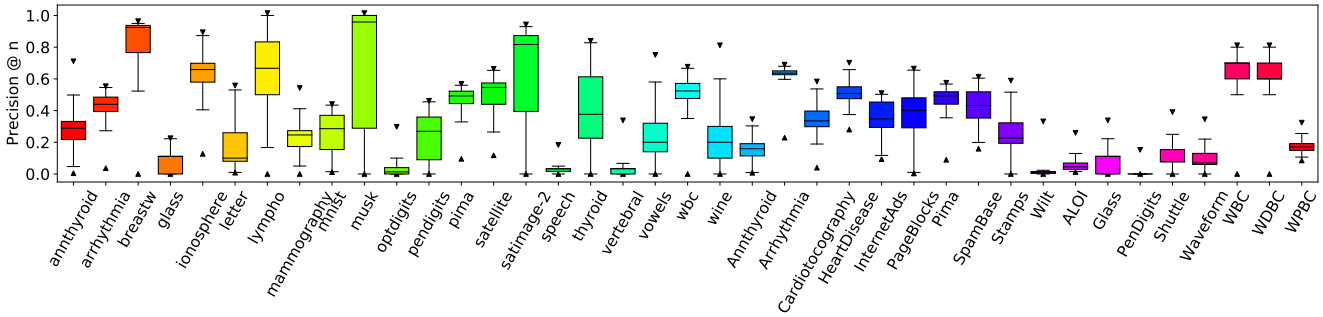

Figure 6: Model performance boxplot (Prec@k) for all datasets, where triangles mark the min and max. Model performance varies significantly for most datasets, showing the importance of model selection.