

# Distribution and volume based scoring for Isolation Forests

1<sup>st</sup> Hichem Dhouib  
*Porsche Digital GmbH*  
 Berlin  
 hichem.dhouib@porsche.digital

2<sup>nd</sup> Alissa Wilms  
*Freie Universität Berlin*  
*Porsche Digital GmbH*  
 Berlin  
 alissa.wilms@porsche.digital

3<sup>rd</sup> Paul Boes  
*Porsche Digital GmbH*  
 Berlin  
 paul.boes@porsche.digital

**Abstract**—We make two contributions to the *Isolation Forest* method for anomaly and outlier detection. The first contribution is an information-theoretically motivated generalisation of the score function that is used to aggregate the scores across random tree estimators. This generalisation allows one to take into account not just the ensemble average across trees but instead the whole distribution. The second contribution is an alternative scoring function at the level of the individual tree estimator, in which we replace the depth-based scoring of the Isolation Forest with one based on hyper-volumes associated to an isolation tree’s leaf nodes.

We motivate the use of both of these methods on generated data and also evaluate them on 34 datasets from the recent and exhaustive “ADBench” benchmark, finding significant improvement over the standard isolation forest for both variants on some datasets and improvement on average across all datasets for one of the two variants. The code to reproduce our results is made available as part of the submission.

## I. INTRODUCTION

Isolation Forest (IF) [8], [10] is one of the most commonly used machine learning methods for anomaly detection, due to its combination of short fitting and evaluation times with high performance across a variety of different types of anomalies. Abstractly, we can view IF as an ensemble-based anomaly detection method, in which a given (and in general unlabeled) training data set  $\mathcal{X} \subset \mathbb{R}^d$  is used to fit an ensemble  $(E_i)_{i=1}^n$  of  $n$  estimators  $E_i$  according to some fitting algorithm that we denote as  $\mathcal{A}$  (and that in general would be non-deterministic). This ensemble is then used to classify any given point  $x \in \mathbb{R}^d$  as outlier or inlier based on the following two steps:

First, an *anomaly score* is obtained for each estimator based on a given *per-estimator scoring function*  $\phi$  such that  $\phi(E_i, x) \in \mathbb{R}$  is the *anomaly score* given to  $x$  by the  $i$ -th estimator.

Second, the individual scores of the estimators are then aggregated into a final score via an *aggregation function*  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  such that the IF algorithm classifies  $x$  as an anomaly if

$$h(\phi(x)) \geq \tau,$$

where  $\tau \in \mathbb{R}$  is some threshold that is specified as part of the input and  $\phi = (\phi_1(x), \phi_2(x), \dots, \phi_n(x))$  with  $\phi_i(x) :=$

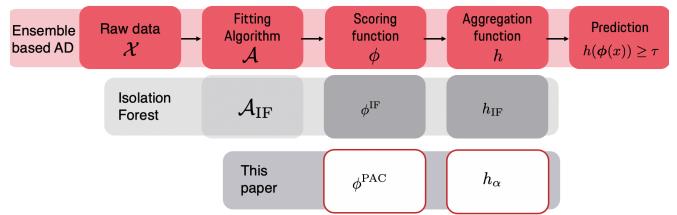


Fig. 1: Overview of our framework. The top red boxes indicate the flowchart of an ensemble based anomaly detection. The grey boxes underneath show the specifics of the Isolation Forest. At the bottom in white with red border are the contributions of this paper visualized.

$\phi(E_i, x)$ . See Fig. 1 for an overview of this process.<sup>1</sup>

We can hence specify any ensemble-based anomaly detection method by a tuple  $(n, \mathcal{A}, \phi, h, \tau)$ . In the specific case of the IF, the fitting algorithm  $\mathcal{A}_{\text{IF}}$  fits a random forest, in which each tree is grown randomly until either a leaf node contains only a single point in a subsampling set  $\mathcal{Y} \subseteq \mathcal{X}$  or a maximal depth is reached. The per-estimator scoring function  $\phi^{\text{IF}}$  is then defined such that

$$\phi^{\text{IF}}(E_i, x) = \frac{d_i(x)}{c_{|\mathcal{Y}|}},$$

where  $d_i(x)$  equals the depth of the leaf node associated to point  $x$  and  $c_{|\mathcal{Y}|}$  is a constant that only depends on the size of the subsampling set used to fit the trees. Finally, the aggregation function  $h$  is defined as<sup>2</sup>

$$h_{\text{IF}}(x) = 2^{-\frac{\sum_{i=1}^n d_i}{n}},$$

that is, the aggregate score is simply the sample mean of the per-estimator scores combined with an outer “normalization” function that ensures the score lies in the interval  $[0, 1]$ , anomalies correspond to points with higher anomaly score, and the

<sup>1</sup>In general, the co-domains of both of these functions could be higher-dimensional, as could be  $\tau$ , with an appropriate ordering. However, for the sake of simplicity, we limit our presentation to real-valued scoring and aggregation functions only.

<sup>2</sup>The authors in [8] refer to  $h_{\text{IF}}$  as the “scoring function”, however, given the distinction between the scoring and the aggregation step that we make in this paper we felt the need to move away from that terminology.

points are more evenly distributed than under the logarithmic scaling of  $\phi^{\text{IF}}$ . In terms of the above tuple notation, IF is then specified as

$$\text{IF} \equiv (n, \mathcal{A}_{\text{IF}}, \phi^{\text{IF}}, h_{\text{IF}}, \tau),$$

where we leave the specification of  $n$  and  $\tau$  implicit.

### A. Contributions

In this paper, we contribute two logically independent variants of the IF method. In particular, while we don't touch the fitting routing  $\mathcal{A}_{\text{IF}}$  (as the core of the IF algorithm), we study a different *per-estimator scoring function*  $\phi^{\text{PAC}}$  that is inspired by computational learning theory and also introduce a parametrized family of aggregation functions  $h_\alpha$  that have  $h_{\text{IF}}$  as a special case. We motivate the use of these variants for anomaly detection by evaluating them both on generated toy datasets, to exemplify their strengths compared to the “plain vanilla” IF, as well as on 34 datasets taken from the recent and exhaustive ADBench benchmark [3]. We find that our the variant based on the family  $h_\alpha$  slightly outperforms the standard IF on average, while the variant based on the scoring function  $\phi^{\text{PAC}}$  performs worse on average but outperforms IF and all other unsupervised anomaly detection methods benchmarked in [3] on 6 out of 34 datasets.

### B. Related Work

1) *Generalised Isolation Forest:* The original Isolation Forest is considered a state-of-the-art anomaly detection algorithm. However, in some cases the limits of this algorithm are reflected in a poor anomaly detection performance. Among the disadvantages is the random choice of the split value used in the construction of the trees. The performance of this algorithm also suffers when it comes to dealing with high-dimensional data or clusters of anomalies. There exist several IF generalisations which tackle the weaknesses of the original Isolation Forest paper. In Deep Isolation Forest [19] a deep neural network architecture is used, in the HDoutliers algorithm [16], extreme value theory is used to calculate a more suitable threshold for the anomalies and in PIDForest [2] a feature selection mechanism is introduced to identify the most relevant features to improve the performance on high dimensional data, which is known as the curse of dimensionality. In Extended Isolation Forest [4] and Generalized IF [6] feature subsetting and categorical feature handling mechanisms, and different distance measures were added to tackle high dimensional datasets and mixed data types. In SCIForest [9] a sliding window approach is introduced to process stream data. In [11] different aggregation function are used to combine the tree scores. In [17] a probabilistic mechanism is used that allows to choose better split values to detect anomalies hidden between clusters more effectively.

2) *Computational learning for anomaly detection:* Many recent works around PAC (Probably Approximately Correct) and anomaly detection have tried to find guarantees for semi-supervised [7] and unsupervised learning [15]. This research direction is motivated by the need for guarantees in order to

bring learning tasks into (safety critical) real world applications like autonomous driving [1], manufacturing [12] and internet of medical things [5]. This work [15] investigates the sample complexity, in the PAC context, of unsupervised anomaly detection. We re-implemented and generalized their PAC motivated algorithm, since their experimental results where not reproducible. This generalization is practically motivated and might not inherit their theoretical proofs.

### C. Code

We make all code required to reproduce the results available.<sup>3</sup>

## II. METHOD

In this section, we introduce the two main contributions of this work, a novel per-estimator score function and a novel aggregation function for ensemble-based anomaly detection methods. As they are logically independent, we introduce them in separate subsections. In each case, we first define them formally and then motivate their usefulness in the context of anomaly detection, using generated datasets. In section III, we'll apply these methods to actual benchmark datasets.

### A. Distribution-based aggregation functions

As discussed in the introduction, an important part of an ensemble-based anomaly detection method is the question how to aggregate the individual per-estimator scores  $\phi_i(x)$  into an aggregate score, in order to classify a point. As we've seen, in IF (a function of) the sample mean is used to aggregate. This is a natural and intuitive choice, however, there can be situations, in which a good estimator will take into consideration features of the vector of estimator scores  $\phi(x)$  other than its sample mean.

For example, consider a dataset  $\mathcal{X} \subset \mathbb{R}^d$  that contains  $N - 1 \gg 1$  inlier points drawn at random from the  $d$ -dimensional unit cube and a single outlier point  $\tilde{x} = (1.05, 0.5, \dots, 0.5)$  that lies just slightly outside of this cube along a single dimension. In this setup, it is possible to separate this point using only a single decision node in a random tree, however, sampling such a tree is statistically relatively unlikely and, as such, most estimators will contain  $\tilde{x}$  in a leaf node at a depth similar to the inlier points. Yet, for large  $N$ , it is, on average, significantly *more* likely for a random tree to separate  $\tilde{x}$  using very few decision nodes than it is for inlier points. As such, we intuitively expect that for sufficiently large  $n$ , there would be significantly more estimators that isolate  $\tilde{x}$  at low depth than there are estimators that isolate any inlier points at that depth. For this reason, we would expect an aggregation function that places disproportionate weight onto high per-estimator scores to lead to a classifier that outperforms one that only considers the sample mean across all estimators.

An extreme version of this idea would be to place all the weight onto the single estimator with the *worst* anomaly score, however, such an extreme weighing mechanism would surely

<sup>3</sup>[https://github.com/porscheofficial/distribution\\_and\\_volume\\_based\\_isolation\\_forest](https://github.com/porscheofficial/distribution_and_volume_based_isolation_forest)

not lead to a robust classifier and more often than not “throw away the baby with the bath water”. Instead, in this paper, we introduce a *family* of aggregation functions, parametrized by a single parameter  $\alpha$ , that lets users “tune” the sensitivity of the aggregation step to estimators with below-average anomaly scores. Specifically, define the functions  $f_\alpha : \mathbb{R}_{\geq 0}^d \rightarrow \mathbb{R}$ , for  $\alpha \in [0, 1) \cup (1, \infty)$ , via the mapping

$$f_\alpha(\mathbf{x}) = n^{\frac{1}{\alpha-1}} \left( \sum_{i=1}^n x_i^{1-\alpha} \right)^{\frac{1}{1-\alpha}}$$

and  $f_\beta = \lim_{\alpha \rightarrow \beta} f_\alpha$  for  $\beta \in \{1, \infty\}$ . This leads to the corresponding aggregation functions for  $\alpha \geq 0$ ,

$$h_\alpha(\mathbf{x}) = 2^{-f_\alpha(\mathbf{x})}$$

In Appendix A, we use a connection of these functions to the *Rényi divergences* from information-theory to show that they satisfy the following properties: satisfy the following properties:

- 1)  $h_0 = h_{\text{IF}}$
- 2)  $h_\infty(\mathbf{x}) = \max(\mathbf{x})$
- 3)  $\alpha \geq \alpha' \Rightarrow h_\alpha(\mathbf{x}) \geq f_{\alpha'}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}_{\geq 0}^d, \alpha \geq 0$

In other words, the aggregation functions  $h_\alpha$  functions are monotonically increasing in  $\alpha$  and interpolate between the standard *IF* aggregation function on the one end ( $\alpha = 0$ ) and the maximum on the other end ( $\alpha = \infty$ ). In this sense, the above aggregation functions introduce a strict generalisation of the original *IF* algorithm and we can write

$$\text{IF}_\alpha \equiv (n, \mathcal{A}_{\text{IF}}, \phi^{\text{IF}}, h_\alpha, \tau),$$

with  $\text{IF} \equiv \text{IF}_0$ .

The monotonicity property of the aggregation functions proves to be very convenient for thinking about the effect of changing the value of  $\alpha$ : All else being equal, increasing  $\alpha$  can only *increase* the number of points a classifier considers anomalous, while conversely reducing it can only *decrease* this number. It is for this reason that we call  $\alpha$  the “sensitivity” of the classifier. As a corollary, for any  $n, \mathcal{A}, \phi, \tau$ , and  $\alpha > \alpha'$ , the anomaly detector

$$AD_\alpha \equiv (n, \mathcal{A}, \phi, h_\alpha, \tau)$$

will have a false negative rate smaller or equal to that of the corresponding detector  $AD_{\alpha'}$  (at the cost of a potentially larger false positive rate).<sup>4</sup>

We believe that the above family of aggregation functions introduces a practically useful degree of freedom for designing an anomaly detection method. Indeed, in Fig. 2 we describe and analyse experiments, based on the motivating toy example above, that show a significant increase in performance for  $\text{IF}_\alpha$  when increasing  $\alpha$ .

<sup>4</sup>As changing the threshold also has the effect of changing the predictions in only one direction, it may seem as if the effect of changing  $\alpha$  can similarly be obtained by changing  $\tau$ . That is, the reader might be under the impression that, continuing the example in the main text, for any value  $\alpha'$ , there would in general exist a value  $\tau'$  such that the predictions of the classifier  $(n, \mathcal{A}, \phi, h_{\alpha'}, \tau')$  coincide with those of  $AD_{\alpha'}$ . This is, however, not the case, as changing  $\alpha$  can lead to a different *relative ordering* of data points wrt their aggregated score, while this is impossible by changing  $\tau$ .

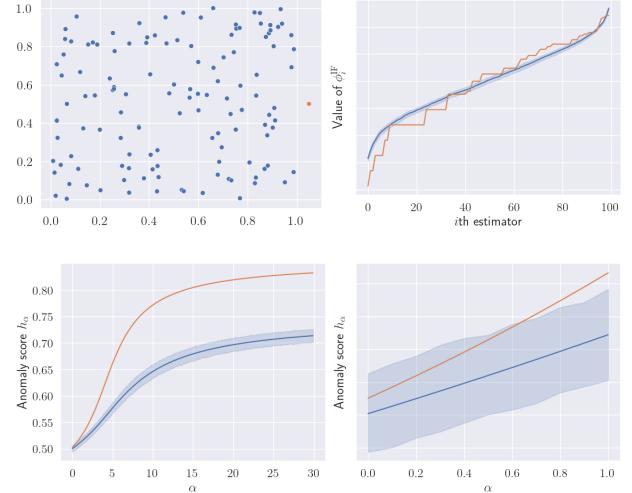


Fig. 2: Motivating the use of distribution-based aggregation functions: We sample a dataset consisting of  $N = 128$  points with 127 inliers (blue) uniformly sampled from a  $d$ -dimensional unit cube and a single anomalous point (orange) slightly outside along a single dimension (*Top left*). Fitting an isolation forest with  $n = 100$  estimators to such a dataset leads to a distribution of (sorted) per-estimator scores in which the anomaly receives a “tail” of significantly worse scores than most inliers (*Top right*). As a result, we can separate the outlier from the inliers by using an aggregation function with sufficiently large  $\alpha$  (*Bottom left*). In particular, zooming into this curve for  $\alpha \leq 1$ , we see that using the sample mean ( $\alpha = 0$ ) leads to an outlier anomaly score better than many inliers (blue shaded regions correspond to the 95th percentile of inlier scores) (*Bottom right*). Running the above experiment a hundred times for  $d = 10$  leads to average AUCROC score of 0.78 for the *IF* algorithm, as opposed to 0.98 when using  $\text{IF}_\infty$ .

### B. Hypervolume-based scoring function

The other important part of an ensemble-based anomaly detection method is the question how to choose the per-estimator scores  $\phi_i(x)$  for a given data point  $x$ .

The standard *IF* algorithm uses random trees as estimators. As such, the leaf nodes  $(L_j^{(i)})_j$  of the  $i$ th estimator  $E_i$  by construction partition the space  $\mathbb{R}^d$  into a set of axis-aligned hyper-rectangles, one for each leaf node. The boundaries of these rectangles simply correspond to the threshold tests at the decision nodes that lead up to a leaf node. With slight abuse of notation, in the following we will use  $L_j^{(i)}$  to denote both this rectangle as well as the leaf node. Moreover, we will use  $L^{(i)}(x)$  to denote that hyperrectangle in the partition corresponding to  $E_i$  that contains the data point  $x$ .

Instead of the classical *IF* scoring based on the depths of the leaf nodes, we propose a scoring function motivated by the Probably Approximately Correct (PAC) framework. The PAC-RPAD framework is a method developed in [15] to show

polynomial sample complexity for learning an unsupervised anomaly detection method with proven guarantees. Inspired by this framework, we define the per-estimator scoring function

$$\phi^{PAC}(E_i, x) = \frac{|L_j^{(i)}(x)|}{|\mathcal{Y}|} \cdot \frac{V(\mathcal{Y})}{V(L_j^{(i)}(x))}, \quad (1)$$

where  $|L_j^{(i)}|$  denotes the number of points of the subsampling set  $\mathcal{Y}$  that are contained in  $L_j^{(i)}$ ,  $V(L)$  denotes the volume of the hyperrectangle  $L$  and  $V(\mathcal{Y})$  denotes some bounding volume that contains all points in  $\mathcal{Y}$  but whose total size is a hyperparameter of the algorithm.

To understand the motivation behind this scoring function, note that the first of the two factors in the RHS of (1) constitutes a density estimate of the likelihood of sampling a given hyper-rectangle when randomly sampling a point in the subsampling set, while the second factor measures the relative volume of this rectangle, which coincides with the probability of sampling this rectangle when sampling a point in the bounding volume uniformly at random. As such, points with bad (i.e. low) anomaly score are those that have density significantly below a uniform. See [15] for a detailed discussion. The heuristic behind this choice of scoring function is that outliers  $x$  are expected to be located inside hyperrectangles which occur with small frequency.

Using the above notation and the concepts from the previous section, we then introduce the family of anomaly detection methods

$$PAC_\alpha \equiv (n, \mathcal{A}_{IF}, \phi^{PAC}, h_\alpha, \tau).$$

Just as in the case of the aggregation functions  $h_\alpha$ , an obvious question is under which circumstances we expect hypervolume-based scoring to outperform depth-based scoring. While we do not have a systematic answer to this question, in the following we want to present a simple example in which we found a significant difference in their performance: Consider a dataset  $\mathcal{X}$  that consists of  $N - 1 \gg 1$  inlier points randomly distributed on the  $d - 1$ -dimensional unit sphere and a single outlier point at the origin. In this case, on average, the largest, in volume, axis-aligned hyper-rectangle that contains only the outlier is significantly larger than that of every inlier point. While a similar statement holds true for the average depth, it is not nearly as pronounced as for the volume. This separation between inliers and outlier increases in  $d$ , hence we expect  $PAC_\alpha$  to outperform  $IF_\alpha$  as the dimension increases. This expectation is confirmed by experiments we ran for the above setup and that show a breakdown of the AUCROC score of  $IF_0$  for  $d > 2$ . See Fig. 3 for details.

### III. EXPERIMENTS

#### A. Setup

We used the recently published ADBench [3] suite of benchmark datasets to compare the performance of the anomaly detection methods  $IF_\alpha$  and  $PAC_\alpha$  introduced in this paper with that of the standard isolation forest and other unsupervised anomaly detection methods, for the values  $\alpha \in \{0, 0.5, 1, 2, \infty\}$ . As common, we used the AUCROC metric

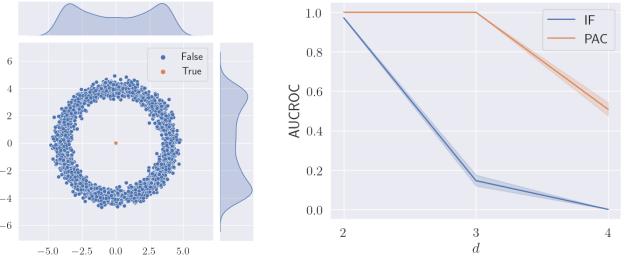


Fig. 3: Motivating the use of a hypervolume-based scoring function: We sample a dataset consisting of  $N = 127$  points uniformly from the surface of a  $d - 1$  dimensional sphere (with some additional Gaussian noise) and place a single anomaly at the origin. The anomaly is easy to separate in the full-dimensional space but “screened” in any lower-dimensional projection of the data (Left). We fit these data to  $IF_0$  and  $PAC_0$  respectively. Running this experiment a hundred times shows a strong and robust separation between the AUCROCs the two algorithms, especially for  $d = 3$ . The reason for the collapse of both algorithms for increasing  $d$  we assume to be an exponentially small likelihood of sampling random trees whose decision nodes imply full-dimensional hyperrectangles (Right).

as the score metric, using the values as stated in [3]. We ran the experiments on 34 datasets, using the code provided in [3] and using a custom implementation of the above methods. For the isolation forest and its variants, we used 100 estimators and evaluated datasets for which an AUCROC value was computed. This means that we ignored datasets for which the AUCROC value was “None” in the ADBench paper.<sup>5</sup>

#### B. Results

We obtained the following results:

- 1) On the ADBench datasets,  $IF_\alpha$  slightly outperforms  $IF$  on average across datasets, for  $\alpha \in \{0.5, 1, 2\}$ , while  $PAC_\alpha$  on average performs significantly poorer (see Fig. 4).
- 2) Consistent with the results of [3], the variance across datasets in the performance of unsupervised anomaly detection methods is high and the algorithms defined herein are no exception to this (see Fig. 5).
- 3) While poor on average, there are datasets in which  $PAC_\alpha$  anomaly detection algorithms significantly outperform their counterpart  $IF_\alpha$  and state-of-the-art methods. To illustrate this, Tab. I in the appendix provides an overview of the performances of  $PAC_\alpha$  against  $IF_\alpha$ .

<sup>5</sup>For reasons not entirely clear to us, we could not reproduce the results for the isolation forest that are stated in [3], even when using the authors’ provided scripts and code without any modifications. Indeed, we found an average of 0.06 AUCROC points across 34 datasets. While this means that the comparison between the IF variants of this paper and the other algorithms should be interpreted with care, it does not affect the comparison between those variants.

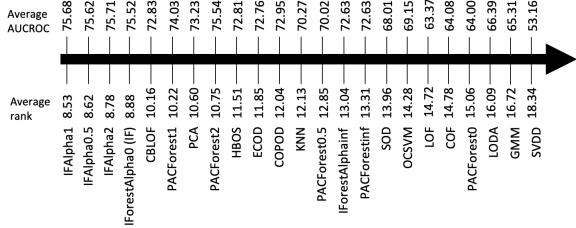
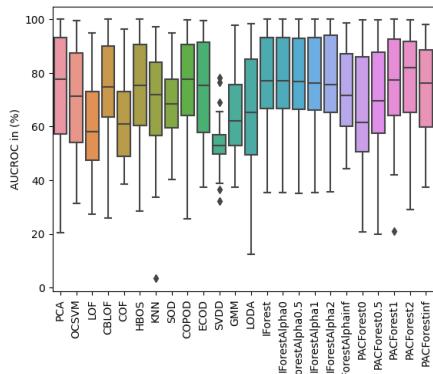


Fig. 4: Ranking of the algorithms according to performance: Following [3], to generate the ranking, we ranked all algorithms on each dataset by AUCROC and then averaged the rank over datasets. For completeness we also state to the average AUCROC across the considered datasets.

and the best of the other comparison algorithms, for a selected subset of the datasets.

Fig. 5: Boxplot of AUCROC IF variants and SOTA algorithms



### C. Discussion

The findings confirm the verdict from [3] that there is not one anomaly detection method that suits all purposes, and the anomaly detection methods discussed in this paper are no exception. At the same time, we see that the introduction of distribution-based scoring has an overall positive, albeit mild, impact on the performance of IF, when  $\alpha$  is relatively small, while performance significantly decreases as  $\alpha \rightarrow \infty$ . This confirms the intuition that *some* sensitivity for outlier votes across estimators helps with anomaly detection, while too much of it does not.

Regarding our volume-based variant of IF, the success hinges very much on the dataset: On some datasets, we found a significant improvement over IF and other unsupervised algorithms, however, on most datasets performance was poor. Unfortunately, at this point we lack a systematic understanding of the criteria that determine whether  $\text{PAC}_\alpha$  is a suitable method for a given dataset, in the absence of ground truth data.

### IV. CONCLUSION AND FUTURE WORK

To conclude, in this note we introduced, motivated and empirically tested two families of ensemble-based anomaly detection methods,  $\text{IF}_\alpha$  and  $\text{PAC}_\alpha$ , both variants of the successful Isolation Forest algorithm. We evaluated them on 34 datasets from the recent and exhaustive “ADBench” benchmark, finding significant improvement over the standard isolation forest for both variants on some datasets and improvement on average across all datasets for  $\text{IF}_\alpha$ . We briefly discussed these results.

For future work, we believe that it could be beneficial to apply the distribution-based aggregation functions  $h_\alpha$  also to other ensemble-based anomaly detection methods, such as LODA [13], or even ensemble-based methods in machine learning beyond anomaly detection. We also believe that it would be valuable to research the possibility of an “auto-tuning” mechanism for  $\alpha$ , that is, some way of tuning  $\alpha$  based on an unlabeled training dataset. Finally, regarding  $\text{PAC}_\alpha$ , we believe that it would be interesting to better understand the types of anomaly datasets on which hypervolume-based scoring is a better choice than depth-based scoring.

### ACKNOWLEDGMENT

We thank Dr. Mahdi Manesh, Regina Kirschner, and the Porsche PANAMERA project for valuable discussions, as well as Prof. Jean-Pierre Seifert and Niklas Pirnay for discussions during the early phase of this project.

### REFERENCES

- [1] Alec Farid, Sushant Veer, Boris Ivanovic, Karen Leung, and Marco Pavone. Task-relevant failure detection for trajectory predictors in autonomous vehicles. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1959–1969. PMLR, 14–18 Dec 2023.
- [2] Parikshit Gopalan, Vatsal Sharan, and Udi Wieder. Pidforest: Anomaly detection via partial identification. 2019.
- [3] Songqiao Han, Xiyang Hu, Hailiang Huang, Mingqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. 2022.
- [4] Sahand Hariri, Matias Carrasco Kind, and Robert J. Brunner. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1479–1489, apr 2021.
- [5] Xiayan Ji, Hyonyoung Choi, Oleg Sokolsky, and Insup Lee. Incremental anomaly detection with guarantee in the internet of medical things. *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation*, 2023.
- [6] Julien Lesoult, Cédric Baudoin, Marc Spigai, and Jean-Yves Tourneret. Generalized isolation forest for anomaly detection. *Pattern Recognition Letters*, 149:109–119, September 2021.
- [7] Shuo Li, Xiayan Ji, Edgar Dobriban, Oleg Sokolsky, and Insup Lee. PAC-wrap. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, aug 2022.
- [8] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. pages 413–422, 2008.
- [9] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. On detecting clustered anomalies using sciforest. 2010.
- [10] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data*, 6:3:1–3:39, 2012.
- [11] Antonella Mensi and Manuele Bicego. Enhanced anomaly scores for isolation forests. *Pattern Recognition*, 120:108115, December 2021.
- [12] Oliver Niggemann, Asmir Vodencarevic, Alexander Maier, Stefan Windmann, and Hans Kleine Büning. A learning anomaly detection algorithm for hybrid manufacturing systems. In *The 24th International Workshop on Principles of Diagnosis (DX-2013)*, Jerusalem, Israel, 2013.

- [13] Tomáš Pevný. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304, 2016.
- [14] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, volume 4, pages 547–562. University of California Press, 1961.
- [15] Md Amran Siddiqui, Alan Fern, Thomas G. Dietterich, and S. Das. Finite sample complexity of rare pattern anomaly detection. 2016.
- [16] Priyanga Dilini Talagala, Rob J. Hyndman, and Kate Smith-Miles. Anomaly detection in high dimensional data. 2019.
- [17] Mikhail Tokovarov and Paweł Karczmarek. A probabilistic generalization of isolation forest. *Information Sciences*, 584:433–449, January 2022.
- [18] Tim van Erven and Peter Harremans. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.
- [19] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–14, 2023.

## APPENDIX

### A. Properties of $f_\alpha$

To clarify the connection of the functions  $f_\alpha$  to information theory and show the claimed properties, let us first define the  $\alpha$ -Rényi divergence for  $\alpha \in (0, 1) \cup (1, \infty)$ , and  $p, q \in \mathbb{R}_{\geq 0}^n$  as

$$R_\alpha(p\|q) = \frac{1}{\alpha - 1} \ln \sum_{i=1}^d p_i^\alpha q_i^{1-\alpha}.$$

with  $R_\beta(p\|q) = \lim_{\alpha \rightarrow \beta} R_\alpha(p\|q)$  for  $\beta \in \{0, 1, \infty\}$ . The Rényi divergences, introduced in [14], generalize the Kullback-Leibler divergence, or relative entropy (which corresponds to  $\alpha = 1$ ) and have a variety of use cases in information [14], [18]. Here, we link them to the aggregation functions introduced in the main text via the simple identity

$$f_\alpha(\mathbf{x}) = \exp\left(-R_\alpha\left(\frac{\mathbf{1}}{n}\middle\|\frac{\mathbf{x}}{n}\right)\right),$$

where  $\mathbf{1}$  is the vector of ones. The claimed properties of  $f_\alpha$  are then a direct consequence of the properties of the Rényi divergences, see e.g. Theorems 3, 4 and 6 in [18].

TABLE I: Detailed results for the AUCROC for SOTA algorithms and IF variants on selected datasets. The highest AUCROC is highlighted in bold. The best value of  $\alpha$  and best SOTA algorithm are given in parentheses behind the AUCROC value.

Dataset	best IF $_{\alpha}$	best PAC $_{\alpha}$	best SOTA
annthyroid	82.12 (0)	<b>91.47</b> (0.5)	82.12 (IForest)
cover	<b>96.58</b> ( $\infty$ )	92.57 ( $\infty$ )	93.73 (PCA)
landsat	48.79 (1)	<b>66.24</b> (0)	63.61 (SVDD)
thyroid	97.84 (0.5)	<b>98.85</b> (0)	97.83 (IForest)
Waveform	69.49 (1)	<b>86.88</b> (0)	75.03 (COPOD)
wine	76.4 (0.5)	<b>92.59</b> (0.5)	91.36 (HBOS)
WPBC	52.66 (0)	<b>63.63</b> (0.5)	52.66 (IForest)