An Empirical Characterization of the Stability of Isolation Forest Results

Alberto Azzari(✉) and Manuele Bicego

University of Verona, Verona, Italy
{alberto.azzari,manuele.bicego}@univr.it

Abstract. Isolation Forests (IForest), a specific variant of Random Forests tailored for anomaly detection, operate by isolating points through recursive partitioning. Despite their widespread use and enhancements in splitting rules, training schemes, and anomaly scoring, an often overlooked aspect is their stability due to the inherent randomness. Surprisingly, most studies and empirical evaluations report results based on a single execution or on the average of a few executions, potentially overlooking significant variability due to this randomness. This paper presents a detailed investigation of the stability of IForests' outcome, proposing some empirical evidence that there may be substantial differences in results across different runs. By exploiting concepts from the field of Ensemble Classifiers, we propose a possible explanation and a strategy to mitigate this instability. Even if we limit our examination to the original IForest model using standard parameters and datasets from the foundational papers, our study underscores the importance of accounting for the random nature of IForests and offers insights and recommendations for practitioners.

Keywords: Isolation Forest · Ensemble Learning · Anomaly Score

1 Introduction

Isolation Forests (IForest) [18,20] represent a particular kind of Random Forests [2] specifically designed for anomaly detection, which usefulness has been shown in many different scenarios [1,6,7]. The main idea behind these methods is that it is possible to measure the isolation of a point by looking at the depth it reaches in a decision tree built on a set of samples of the considered problem. Points that are well inside the distribution would need more splits to be isolated, i.e., would reach a deeper leave in the tree; on the contrary, an outlier, being separated by definition, will be isolated in few splits, i.e., it would reach a leaf in fewer steps. Since their introduction in 2008, isolation forests have received increasing attention from researchers and have been extended in different directions. A non-exhaustive list includes improvements in the splitting rule [11], in the training scheme [10,14], in the computation of the anomaly score [19,21] or even in the applicability to non-vectorial data [22,27].

A. Torsello et al. (Eds.): S+SSPR 2024, LNCS 15444, pp. 166–176, 2025.
https://doi.org/10.1007/978-3-031-80507-3_17

One of the crucial ingredients of IForests is represented by its intrinsic random nature. In addition to being a Random Forest (thus inheriting the randomicity intrinsic in the bootstrapping procedure), these methods are based on Extremely Randomized Trees (ERT - [9]), i.e., Decision Trees in which randomization is driven to the extreme. Assuming the general scenario of a node's splitting rule defined with a threshold on a feature, within ERT, the rule is defined i) by randomly choosing the feature and ii) by randomly selecting the threshold in the domain of such feature. With this extreme randomization, one may argue about the stability of the results of these methods in practical scenarios. Surprisingly, this aspect is rather underestimated in the literature. In the original paper of Isolation Forests [18], there is no citation to this aspect, and reported empirical results are obtained with only one run of the method. Similarly, classical surveys on anomaly detection, which include Isolation Forests, such as [6] or [7], do not consider this aspect, presenting results of a single run, and the same holds for more recent surveys in specific application scenarios, such as [1] (industrial manufacturing), [13] (natural gas), or [8] (intrusion detection). When this aspect is taken into account (e.g., the journal version of Isolation Forests [20] or very recent approaches – e.g., [26]), the solution is to repeat 10-15 times the execution and take the average, without explicitly examining the specific issue.

This paper is aimed at explicitly studying this aspect, namely at investigating the stability of the results of Isolation Forests. To the best of our knowledge, the only other paper[1] that explicitly considers this aspect is the very recent [4]. In such a paper, the authors conducted a quite large study on the impact on the results of different aspects of IForests, such as parameters (sample size, number of trees), the presence of outliers in the training set, the dimension of the data, and others. Crucially, they also analyze the "Randomness effect", i.e., the stability of IForest results due to the high randomization present in the model. To do this, the authors checked the standard deviation of the accuracy on 10 repetitions of IForest. The analysis, based on a single dataset, led to the conclusions that *"These results illustrate the stability of Isolation Forest despite its randomness. We can, therefore, rely on a single execution of Isolation Forest"*. We will show here that such a conclusion is rather superficial, showing that results among different runs may be rather different. We will also provide an interpretation of this instability by reasoning on the specific way the anomaly score is computed from the forest, borrowing concepts and ideas from the Ensemble Classifier field; following such view, we will also propose a way to reduce such impact partially. In our study, we focus on the original version of IForest, [18,20], employing the standard parametrization and the datasets used in those papers. Even if we do not provide a thorough analysis of different extensions of IF and different datasets, we are convinced that our results provide clear evidence that considering the random nature of IForests is definitely crucial.

[1] Some recent studies (e.g. [25]) tackled the problem of stability of classic Random Forests; even these results are of general interest, they are thought for the classical classification/regression training schemes, and thus do not apply to Isolation Forests, which are based on a very peculiar (and highly randomized) training scheme (ERT).

The rest of the paper is organized as follows. In Sect. 2, we will briefly summarize Isolation Forests, mainly to set up the notation. In Sect. 3, we present the empirical evidence about stability, providing a possible interpretation and (partial) solution in Sect. 4. Finally, in Sect. 5, we conclude our paper.

2 Background

The Isolation Forest (IForest) [18,20] represents an ensemble of Isolation Trees (iTrees). An iTree is a proper binary tree where each node has exactly zero (a leaf) or two (an internal node) children nodes. Each internal node is equipped with a test, which allows an object to go to the left or to the right child according to the test result. Like in classical classification and regression decision trees [3], a test t on an object $\mathbf{x} = x_1, \cdots x_m$ is defined by a tuple $t = (f, v)$ so that the object $\mathbf{x}$ follows the left branch if $\mathbf{x}_f < v$, the right one otherwise, with $\mathbf{x}_f$ being the value of the feature f in the object $\mathbf{x}$.

Training Isolation Trees and Forest. To train an Isolation Tree, we follow a classical top-down, recursive strategy. We start with the whole training set $D = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(L)}\}$ composed by L objects, and choose a test $t = (f, v)$ for the root. The test splits D into D_L and D_R, according to the answer of the test. The sets D_L and D_R are then used to determine the tests in the left and right children. This is recursively done in each node until the tree reaches a height limit or a node contains a minimum number of samples. To select the test $t = (f, v)$ within a node, Isolation Trees use the Extremely Randomized Trees (ERT—[9]) paradigm. Specifically, given a set $D = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(L_k)}\}$ of objects reaching a node, we randomly select a feature f and then randomly generate a split threshold v within the range (f_{min}, f_{max}). Here, f_{min} and f_{max} represent the minimum and maximum values of the feature f among the objects in D, defined as:

$$f_{min} = \min_{i=1\cdots L_k} \mathbf{x}_f^{(i)}, \quad f_{max} = \max_{i=1\cdots L_k} \mathbf{x}_f^{(i)}$$

where $\mathbf{x}_f^{(i)}$ is the value of feature f for the object $\mathbf{x}^{(i)}$.

The Isolation Forest (IForest) represents an ensemble of iTrees. Following the classical recipe of Random Forests [2], each iTree is created, with the procedure described in the previous subsection, starting from a random sub-sampling of the training set.

Computing the Anomaly Score. Given a trained Isolation Forest, the anomaly score $a_{IF}(\mathbf{x})$ of an object $\mathbf{x}$ is computed as follows. First of all, the average path length $\bar{h}(\mathbf{x})$ of $\mathbf{x}$ over multiple trees is computed:

$$\bar{h}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} h_t(\mathbf{x})$$

where T is the number of trees in the forest, and $h_t(\mathbf{x})$ is the number of edges along the path $\mathbf{x}$ follows in the t-th tree. The anomaly score $a_{IF}(\mathbf{x})$ for the object $\mathbf{x}$ is then computed as:

$$a_{IF}(\mathbf{x}) = 2^{-\frac{\bar{h}(\mathbf{x})}{c(n)}} \tag{1}$$

where $c(n)$ is a normalization coefficient representing the expected path length of unsuccessful searches in Binary Search Trees, derived from Isolation Forest [18] and computed with the following equation:

$$c(n) = 2\,H(n-1) - \left(\frac{2(n-1)}{n}\right)$$

where $H(i)$ is the i-th harmonic number ($H(i) = \ln(i) + 0.5772156649$ **(Euler's constant)**) and n is the number of instances in the given dataset.

3 Stability of Isolation Forest Results

In this section, we analyze the stability of the results of the IForest method; as in many outlier detection problems, to quantify the performances we employed the ROC-AUC measure, computed starting from the anomaly scores defined in Eq. (1). We focus here on the standard configuration of IForest, suggested in the original papers [18,20] and adopted in many other settings (e.g., [4]); in particular each Isolation Tree is built with a random subsample of 256 objects, growing the three until the maximum depth is 8 (i.e. $\log(n)$, $n = 256$). The forest is composed of 100 trees. Our empirical analysis is based on the datasets employed in the original papers of Isolation Forests [18,20] – except for the Annthyroid problem, for which we were not able to recover the same version (in the web version we found[2] the dataset contains 7200 objects, whereas in the Isolation Forest paper 6832, without citation to the preprocessing). A summary of the datasets is reported in Table 1.

Table 1. Datasets employed in the experiments.

Name	Number of objects	Number of features	Percentage of outliers
Http	567497	3	0.4%
Cover	286048	10	0.9%
Mulcross	262144	4	10%
Smtp	95156	3	0.03%
Shuttle	49097	9	7%
Mammography	11183	6	2%
Annthyroid	7200	6	2%
Satellite	6435	36	32%
Pima	768	8	35%
Breastw	683	9	35%
Arrhythmia	452	274	15%
Ionosphere	351	32	36%

[2] https://odds.cs.stonybrook.edu/annthyroid-dataset/.

To analyse the stability, we tried to answer to the following practical question: suppose that I get one result with IForest, how different would be the result if I make another run? In order to quantify this, we adopted the following strategy:

1. Train the first IForest, compute the anomaly scores and determine the AUC (AUC 1^st run).
2. Train another IForest, computing again the anomaly scores and the AUC.
3. compare the two AUCs and two sets of anomaly scores to assess if there is a statistically significant difference among them. To do that, we exploited the equivalence between the AUC and the Wilcoxon or Mann-Whitney U test statistic, as explained in [12]. The employed test also returns a p-value[3]
4. Repeat the above procedure 200 times, counting the number of times the two runs were considered different with a statistical significance; as significance level, we employed $\alpha = 0.05$, also performing a Bonferroni Correction for multiple tests. In order to reduce the computational burden of training so many IForests, we created 2000 Isolation Trees, randomly sampling from this superset the 100 trees of each Forest.

3.1 Results

The obtained results are shown in Table 2 for the different datasets: in particular, we reported the number of times, over the total, it happened that a run of IForest was different than the first run with a statistical significance (Num. S.S.D.). To have a clearer idea, we also reported the AUC of the first run and the averaged AUC of the 200 other runs (with min and max).

Table 2. Results for standard configuration of IForest.

Problem	Num. S.S.D.	AUC 1^{st} run	Avg AUC other runs (min-max)
Http	169/200	1.000	0.999 (0.997–1.000)
Cover	176/200	0.886	0.886 (0.796–0.949)
Mulcross	197/200	0.947	0.956 (0.924–0.978)
Smtp	12/200	0.906	0.907 (0.892–0.925)
Shuttle	180/200	0.996	0.997 (0.995–0.999)
Mammography	75/200	0.861	0.862 (0.842–0.886)
Annthyroid	161/200	0.832	0.816 (0.776–0.865)
Satellite	127/200	0.685	0.697 (0.669–0.746)
Pima	20/200	0.669	0.675 (0.642–0.706)
Breastw	0/200	0.989	0.986 (0.982–0.990)
Arrhythmia	0/200	0.800	0.804 (0.777–0.832)
Ionosphere	1/200	0.855	0.854 (0.839–0.866)

[3] Code is available here: https://github.com/alistairewj/auroc-matlab.

The most evident observation that can be derived from the table is that for many different datasets, the number of statistically different runs is very high, being almost 0 in only 3 cases. Actually, if we consider the AUC, we can observe that within the 200 runs, values are rather different than the AUC of the first run (for example, for the Cover dataset, the minimum is 0.09 lower than the first run). From these numbers, it seems evident that the randomness effect should be carefully considered and that we cannot show the results deriving from a single run of IForest. The phenomenon is more evident for large datasets, whereas it is less relevant for datasets with less than 1000 objects. This seems reasonable since with smaller datasets, the sets of objects used to train each tree (i.e., via subsampling) are more overlapped, thus resulting in fewer diverse trees.

3.2 Results with More Trees

Results presented in the previous section clearly show that randomicity in IForests can lead to unstable results. One possible (and rather obvious) solution would be to increase the number of trees, thus exploiting the ensemble effect to reduce variability. This option would clearly increase the computational complexity, but we are interested here in a different question: does this solution increase the stability in all cases? To investigate this aspect, we repeated the experiments of the previous section by increasing the number of trees, i.e., by considering 200, 400, and 800 trees. We aggregate the results by taking averages over groups of datasets in order to understand the behavior when considering the following specific aspects: i) size of the dataset (small vs. large problems), ii) dimensionality of the feature space (high dimensional vs. low dimensional problems), iii) AUC 1^{st} run (difficult vs. easy problems) and iv) percentage of outliers (problems with few or many outliers). For each aspect, we split the 12 datasets into two groups of 6: for example, for the size, we considered in one group the 6 smallest datasets (Ionosphere, Arrhythmia, Breastw, Pima, Satellite, and Annthyroid), whereas in the other group, the remaining ones. We then took the averages of the number of statistically significant different runs for an increasing number of trees, reporting such values in Fig. 1 for the 4 different aspects.

From the figures, we can observe that, as expected, increasing the number of trees reduces the instability problem in many cases; however, problems may still appear when working with difficult problems involving many outliers.

4 A Possible Explanation

In this section, we provide some considerations on the possible origin of the instability of the IForest results. In particular, we start from the definition of the anomaly score computed from the forest, as given in Eq. (1), and we interpret it using concepts and ideas of the Ensemble Classifier field [17]. Specifically, it is easy to see that Eq. (1) can be rewritten as follows (for the sake of readability,

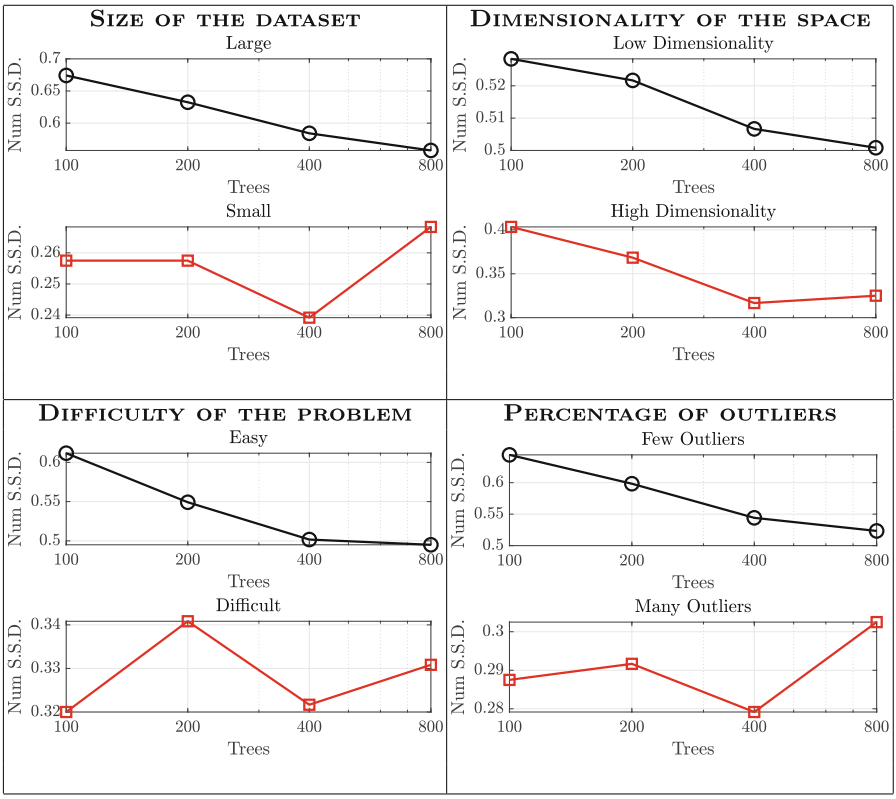

Fig. 1. Results when growing the number of trees.

we removed the $c(n)$ factor, which is constant over different trees, and does not impact our reasoning):

$$a_{\text{IF}}(x) = 2^{-\bar{h}(x)} = 2^{-\frac{1}{T}\sum_{t=1}^{T} h_t(x)} = \prod_{t=1}^{T} 2^{-\frac{h_t(x)}{T}} \tag{2}$$

By considering that each isolation tree t returns its own anomaly score:

$$a_{\text{IT}}(x) = 2^{-\frac{h_t(x)}{T}} \tag{3}$$

we can observe that the original anomaly score $a_{\text{IF}}(x)$ simply represents the output of an ensemble of classifiers, in which the individual outputs are combined together with the PROD rule, one of the combination rules which go under the name of nontrainable (or fixed) combiners [17]. In the ensemble learning community, the PROD rule has been largely studied, both from a theoretical [15,16] and an empirical perspective [23,24], especially in comparison with the SUM rule. From all these studies, it emerges that the PROD rule is rather a severe combiner rule, being a proper choice only in the absence of estimation errors, but

being more sensitive if such errors are present; in such cases, more benevolent rules (such as the SUM rule) represent a more appropriate choice. In particular, authors of [16] suggested that the PROD rule may be less accurate since a single bad score can change the overall output due to the restricted behavior of the product. This means that, in the IForest case, one single bad tree may drastically change the final output; since trees are randomly created, bad trees may be generated.

This possible explanation of the instability of IForests opens the route to the investigation of alternative combiners, which may reduce the instability of IForest. We propose here some preliminary experiments along this direction, employing some classic rules, such as the SUM[4], the MEDIAN, the MIN, the MAX and the TRIMMEDSUM (i.e., the sum rule in which the 5% larger and smaller scores are removed [17]). Results are shown in Table 3, in which we reported the Num. S.S.D. over three datasets: Shuttle, Mammography, and Pima, belonging to different stability category (low, medium, high, respectively).

Table 3. Comparison with other combiners, showing Shuttle, Mammography and Pima datasets.

Method	Shuttle	Mammography	Pima
Original (PROD)	180/200	75/200	20/200
SUM	150/200	50/200	0/200
TRIMMED-SUM	150/200	48/200	4/200
MEDIAN	140/200	64/200	14/200
MIN	200/200	80/200	30/200
MAX	158/200	34/200	0/200

From the Table 3, it seems evident that alternative rules may improve the stability of IForest results, this being especially true for the SUM rule. The only exception is the MIN rule, which, however, can be considered as an approximation of the PROD rule [15]. To include the AUC in the perspective, we report in Fig. 2 a critical diagram showing the result of a Friedman test, followed by the post-hoc Nemenyi test [5], which compares the AUC of the different aggregation rules over the different datasets: the plot reports rankings – the lower the better–, with the difference between methods connected by a line being not statistically different. We can observe that the performances of most of the alternatives are equivalent to those of the original score, except for the MIN and the MAX rule, which results are drastically lower. Thus we can confirm the general findings of Kuncheva, who, in her book [17] says that *"The current understanding is that*

[4] Please note that a novel anomaly score for IForest, which can be seen as equivalent to the SUM rule, has also been investigated in the recent [21]; in such paper, however, it was introduced mainly from a probabilistic perspective, with experiments focused only on improving the AUC.

the average (i.e., the sum), in general, may be less accurate than the product for some problems, but is the most stable of the two".

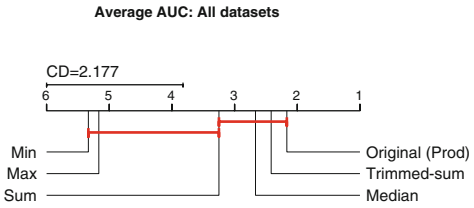

Fig. 2. Critical difference diagram showing the ranks of the employed combiners based on the average AUC across all datasets.

5 Conclusions

This study delves into the stability of Isolation Forests, a widely used anomaly detection technique, with a focus on understanding how its inherent randomness affects the repeatability of its results. Our empirical analysis reveals significant variability in the Area Under the Curve (AUC) scores across multiple runs of the IForest algorithm. This highlights a critical issue: despite their popularity and effectiveness, the random nature of IForest can lead to unstable outcomes, which can undermine the reliability of conclusions drawn from a single execution. Increasing the number of trees can reduce this variability, but doesn't entirely eliminate it. Alternative methods for combining anomaly scores, like the sum, offer improved stability over the default product rule, suggesting that these should be considered to enhance the reliability of IForest outcomes.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Barbariol, T., Chiara, F.D., Marcato, D., Susto, G.A.: A review of tree-based approaches for anomaly detection. In: Tran, K.P. (ed.) Control Charts and Machine Learning for Anomaly Detection in Manufacturing. SSRE, pp. 149–185. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-83819-5_7
2. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
3. Breiman, L., Friedman, J.H., Olshen, R., Stone, C.J.: Classification and Regression Trees. Wadsworth (1984)
4. Chabchoub, Y., Togbe, M.U., Boly, A., Chiky, R.: An in-depth study and improvement of isolation forest. IEEE Access **10**, 10219–10237 (2022)
5. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)

6. Domingues, R., Filippone, M., Michiardi, P., Zouaoui, J.: A comparative evaluation of outlier detection algorithms: experiments and analyses. Pattern Recogn. **74**, 406–421 (2018)
7. Emmott, A.F., Das, S., Dietterich, T., Fern, A., Wong, W.K.: Systematic construction of anomaly detection benchmarks from real data. In: Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, pp. 16–21 (2013)
8. Falcão, F., et al.: Quantitative comparison of unsupervised anomaly detection algorithms for intrusion detection. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 318–327 (2019)
9. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**, 3–42 (2006)
10. Guha, S., Mishra, N., Roy, G., Schrijvers, O.: Robust random cut forest based anomaly detection on streams. In: International Conference on Machine Learning, pp. 2712–2721. PMLR (2016)
11. Hariri, S., Kind, M.C., Brunner, R.J.: Extended isolation forest. IEEE Trans. Knowl. Data Eng. **33**(4), 1479–1489 (2019)
12. Hernández-Orallo, J., Flach, P., Ferri Ramírez, C.: A unified view of performance metrics: translating threshold choice into expected classification loss. J. Mach. Learn. Res. **13**, 2813–2869 (2012)
13. Jha, H., Khanal, A., Seikh, H., Lee, W.: A comparative study on outlier detection techniques for noisy production data from unconventional shale reservoirs. J. Nat. Gas Sci. Eng. **105**, 104720 (2022)
14. Karczmarek, P., Kiersztyn, A., Pedrycz, W., Al, E.: K-means-based isolation forest. Knowl.-Based Syst. **195**, 105659 (2020)
15. Kittler, J.: Combining classifiers: a theoretical framework. Pattern Anal. Appl. **1**, 18–27 (1998)
16. Kittler, J., Hatef, M., Duin, R.P., Matas, J.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. **20**(3), 226–239 (1998)
17. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley (2004)
18. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422. IEEE (2008)
19. Liu, F.T., Ting, K.M., Zhou, Z.-H.: On detecting clustered anomalies using SCiForest. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) ECML PKDD 2010. LNCS (LNAI), vol. 6322, pp. 274–290. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15883-4_18
20. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data (TKDD) **6**(1), 1–39 (2012)
21. Mensi, A., Bicego, M.: Enhanced anomaly scores for isolation forests. Pattern Recogn. **120**, 108115 (2021)
22. Mensi, A., Tax, D.M., Bicego, M.: Detecting outliers from pairwise proximities: proximity isolation forests. Pattern Recogn. **138**, 109334 (2023)
23. Tax, D.M., Duin, R.P., Van Breukelen, M.: Comparison between product and mean classifier combination rules. In: Proceedings of Workshop on Statistical Pattern Recognition, Prague, Czech, p. 39. Citeseer (1997)
24. Tax, D.M., Van Breukelen, M., Duin, R.P., Kittler, J.: Combining multiple classifiers by averaging or by multiplying? Pattern Recogn. **33**(9), 1475–1485 (2000)
25. Wang, Y., Wu, H., Nettleton, D.: Stability of random forests and coverage of random-forest prediction intervals. In: Advances in Neural Information Processing Systems, vol. 36 (2024)

26. Xiang, H., et al.: Optiforest: optimal isolation forest for anomaly detection. In: Proceedings of International Joint Conference on Artificial Intelligence (2023)
27. Zhang, X., et al.: Lshiforest: a generic framework for fast tree isolation based ensemble anomaly analysis. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), pp. 983–994. IEEE (2017)