



# A Novel Anomaly Score for Isolation Forests

Antonella Mensi<sup>(✉)</sup> and Manuele Bicego

Department of Computer Science, University of Verona, Verona, Italy  
{antonella.mensi,manuele.bicego}@univr.it

**Abstract.** Isolation Forests represent a recent variant of Random Forests, specifically designed for one-class classification problems. In the original version, this method builds a set of extremely randomized trees to describe the set of points, subsequently measuring the “anomaly” of a testing point by looking at how much deep it arrives in each tree. Even if few extensions have been recently proposed – mainly aimed at improving the training stage – in most cases the anomaly score is still kept in its original formulation, which does not completely exploit all the information contained in the trained forest. This paper is focused on improving this aspect, and proposes a new approach for the computation of the anomaly score, which exploits the different information linked to the different nodes of the trees of the forest. We investigate three different variants of the novel anomaly score, evaluating them with twelve UCI benchmark datasets, with encouraging results.

**Keywords:** One-class classification · Random Forests ·  
Outlier Detection · Isolation Forests

## 1 Introduction

Random Forests [5] represent a widely used tool for classification and regression, based on creating an ensemble of randomized decision trees [4], where each tree is built on a random subsample of the data and of the features. Randomness is crucial to get diverse trees, reducing the risk of overfitting and the computational complexity. The obtained ensemble method is more robust and performs better than a single tree [5, 18]; actually it has been shown that these tools perform very well in many different fields such as computer vision [3], bioinformatics [6], remote sensing [23] and others, reaching performances which are comparable with other state-of-the-art techniques such as Support Vector Machines and Neural Networks [12].

Even if Random Forests have been mainly used for classification and regression, there also exist some random forest-based approaches for alternative learning paradigms, such as clustering [1, 21, 26, 27, 30], survival analysis [16], ranking problems [7], multi-label classification [17] and one-class classification [8, 14, 15, 20, 26]. In this paper we focus on this latter class, i.e. one-class classification [22], a learning problem in which only objects from one class are available

(the target, or positive, class), and where the aim is to identify whether new objects belong to that class or not [22, 29]. The objects that do not belong to the positive class are also known as outliers or anomalies.

Even if the exploitation of Random Forests in the one-class classification field has not been studied as extensively as for classification and regression, some interesting approaches have been proposed, which can be subdivided mainly into two classes. The first class includes all those methods, such as [8, 26], that solve the one-classification task by creating a synthetic negative class (the outliers), so that a classic classification random forest can be trained. The outlier generating process is often based on assuming a well-defined distribution: one possibility is to sample outliers uniformly on the domain space or to locate them in sparse, isolated regions that contain few inliers [26]. The main advantage of this class of approaches is that standard classification forests can be used without any modification. At the same time these methods can arise some issues: the most important is that the choice of the sampling technique is crucial. For example in a high-dimensional space if we assume outliers to be uniformly distributed, we have to generate a very big number of points to populate the space, and this is often not feasible. In addition, given a specific problem, the chosen distribution may not truly reflect how the outliers would distribute.

The second class of approaches are those based on Isolation Forests, a particular kind of Random Forests introduced by Liu and colleagues in [19, 20]. Within these tools, the goal is not to discriminate objects of different classes but rather to isolate instances, that is to separate one object from the remaining ones. To do that, Isolation Forests partition the data through random and recursive splits along feature axes: a point is isolated when the leaf containing that point is created. Outliers, which are very different in terms of feature values and number, are likely to be separated earlier in the tree building process than inliers. Therefore, to quantitatively measure how much an object is isolated, the authors of [19, 20] propose a scoring function, called *anomaly score*, which is inversely proportional to the length of the path in a tree that the object traverses to reach its leaf, averaged along all trees. As said before, the defined score will be higher for outliers since they are likely to be separated closer to the root. Isolation Forests present many advantages: they can work with only positive instances –and therefore no outliers need to be artificially generated– and they are computationally efficient thanks to the random split mechanism.

Even if Isolation Forests have been shown to be very effective for one-class classification –e.g authors of [11] empirically demonstrated that they are the best existing method to solve one-class classification tasks–, streaming data [13] and clustering [1], their full potential has not yet been completely exploited, especially for what concerns the testing phase. In almost all works dealing with Isolation Forests (see for example [9, 11, 28] or the extension proposed by [14]) the anomaly score is still kept in its original formulation of [19, 20], which does not completely exploit all the information contained in the trained forest<sup>1</sup>. More

---

<sup>1</sup> For the sake of completeness, please note that a new scoring function has been proposed in [15]. This measure, however, is specifically designed for streaming data.

in detail, the scoring function is based on the length of the path traversed by an object (the shorter the path, the more isolated the point). Even if being very reasonable, this measure does not consider the information carried by each node, i.e. it does not consider that not all nodes are equally important in the path—for example a node with few points, e.g. a leaf, is usually more descriptive of the feature space than a bigger node, such as the root. In this paper we overcome this drawback, and propose an extension of the anomaly score: the novel score, which we called *path-weighted anomaly score*, is based on the estimation of a *weighted path length*, which exploits and takes into account the importance of the different nodes of the trees. We designed three different variants of the score, which consider different ways of measuring the “importance” of a node in a path. It is important to note that node weights are computed on training data and therefore, to not increase the testing procedure complexity, we compute them while building the tree, i.e. during the training phase.

The proposed schemes have been evaluated on 12 UCI benchmark datasets for one-class classification [10]. We investigated different parametrizations and configurations, comparing the proposed approach with the standard counterpart: the obtained results are very promising. The rest of the paper is organized as follows: in Sect. 2 we explain in detail the Isolation Forests, while in Sect. 3 we thoroughly define the proposed methodology. Section 4 is dedicated to the experimental part and the related results. Finally, Sect. 5 contains some conclusions.

## 2 Isolation Forests

Isolation Forests are variants of Random Forests introduced by Liu in [19, 20]. The basic idea behind these methods is that one-class classification can be solved via *isolation*, that is by separating one object from the rest of the data, without focusing on discriminating objects of different classes.

To encode the concept of isolation the authors in [19, 20] propose a new tree structure, called *ITree*. The ITree is based on the *Extra-Trees* proposed in [13]. These tools introduce different levels of randomness in the tree construction: for example, instead of evaluating at each node every possible split on a subset of features (as in standard decision trees), Extra Trees select a random split for each feature in the subset. The ITree exploits the extreme version of the Extra-Trees, called *totally randomized trees*, in which every split is completely random (at every node, a random feature is extracted, and a random threshold in the feature domain is chosen). Clearly, ITrees can be built using data coming from only one class. Very recently, some authors [14, 15] investigated alternative approaches to build Isolation Forests: in particular in [14] they develop a function able to evaluate every possible split in a one-class context, while in [15] they design a new criteria which chooses the feature to split on randomly but proportionally to the feature relevance.

To recover the isolation capability of an object, an anomaly score is defined on the basis of the length of the path that the object traverses from the root to its leaf. This measures the number of partitions needed to separate it from the

rest [20]: if an object is found closer to the root, i.e. its path is short, it means that it is easier to separate it, and thus to isolate it from the rest, with respect to objects that end up in deeper leaves. More in detail the anomaly score of an object  $x$  with respect to an Isolation Forest  $\mathcal{F}$  is the following –in the paper we consider the dependence of the score on  $\mathcal{F}$  implicit in order not to make the notation too heavy–:

$$s(x, N) = 2^{-\frac{E(h(x))}{c(N)}} \quad (1)$$

where  $N$  is the number of samples used to train each tree of the forest,  $E(h(x))$  is the average path length across all trees (see below) and  $c(N)$  is a normalization factor, needed to compare trees built on sets of different sizes. To estimate  $c(N)$ , which can be seen as the average path length, we can use the estimation of the average path length of unsuccessful searches in Binary Search Trees [19, 20], which is defined in the following way according to [24]

$$c(N) = \begin{cases} 2H(N-1) - 2(N-1)/N & \text{if } N > 2 \\ 1 & \text{if } N = 2 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where  $H(i)$  stands for the harmonic number. The term  $E(h(x))$  in formula (1) is computed as:

$$E(h(x)) = \frac{\sum_{t \in \mathcal{F}} h_t(x) + \sum_{t \in \mathcal{F}} c(|l_t(x)|)}{|\mathcal{F}|}. \quad (3)$$

where  $t$  is a tree,  $c(|l_t(x)|)$  is a normalization factor needed when  $t$  is not fully grown (which estimates the average depth of the tree which can be built from  $l_t(x)$ ) and  $h_t(x) = |\mathcal{P}_t(x)|$  with  $\mathcal{P}_t(x)$  being the path of  $x$ , i.e. the set of nodes visited by  $x$  from the root to the leaf containing  $x$ . From formula (1) it can be inferred that the score of an object  $x$  is proportional to the inverse of its average path length in the forest: if  $x$  ends up in leaves that are very deep in the trees, its score will be quite low (close to 0), if instead its path ends very early the score will be high (close to 1).

The anomaly score defined in (1) represents a reasonable way to characterize outliers, and thus to solve the one-class classification problem: actually outliers are usually very heterogeneous and low in number with respect to inliers, and they do not follow a predefined distribution. When building an Isolation Forest, they will be more likely separated from the rest of the data very quickly, i.e. after few partitions. In other words, outliers will be likely to traverse a shorter path with respect to inliers, producing an higher anomaly score (usually  $\geq 0.5$  as stated in [20]).

### 3 Proposed Methodology

In this section we describe the proposed approach. The starting observation is that the anomaly score considers each node visited in a path to have the same importance. In this sense, the path length  $h_t(x)$  of a  $x$  in a tree  $t$  can be written as

$$h_t(x) = \sum_{k \in \mathcal{P}_t(x)} 1. \quad (4)$$

The main idea behind the proposed approach is to define a novel anomaly score where nodes in the path are given a weight, which corresponds to specific information that can be retrieved in the forest. The novel scoring function is called *path-weighted anomaly score* and is based on re-defining  $h_t(x)$  as follows.

Given a tree  $t$  and the path  $\mathcal{P}_t(x)$  of an object  $x$ ,  $h_t(x)$  is defined as:

$$h_t(x) = \sum_{k \in \mathcal{P}_t(x)} w_{tk} \quad (5)$$

where  $w_{tk}$  represents the weight the node  $k$  has in the tree  $t$ . Clearly, when considering  $w_{tk} = 1 \forall t, k$ , we have the original anomaly score. The weights  $w_{tk}$  can be defined in several ways, here we investigated three different versions, presented in the following.

### 3.1 Variant 1 – Neighborhood

The first variant is based on the concept of neighborhood defined in [30]: considering a node  $k$  and an object  $x$  in that node, the neighborhood of  $x$  is defined as the set of all the other objects that would pass from  $k$ . More in general, we can define the neighborhood  $N_{tk}$  of a node  $k$  of a tree  $t$  as the set of the objects of the dataset that would pass by  $k$  in their path from the root to the leaves of the tree  $t$ . Clearly the neighborhood of the root is the whole dataset, whereas the neighborhood of a leaf contains only few points. To define the weight, we observe that a node which has a very small and restrictive neighborhood is more important than a larger one, since it is more specific for the object under analysis. In particular, in an Isolation Forest a small neighborhood occurs when we are very deep in the tree (since the number of objects decreases from the root to the leaves) or we are high in the tree and there is an outlier that has been isolated after few partitions (i.e. we have leaves at small depths).

We thus want to give more weight to nodes with a smaller neighborhood, which leads to the following definition of  $w_{tk}^N$ . Given a tree  $t$  and a node  $k$  in  $t$ , its weight  $w_{tk}^N$  is:

$$w_{tk}^N = \frac{1}{|N_{tk}|} \quad (6)$$

where  $N_{tk}$  is the neighborhood of the node  $k$ , i.e. the set of points passing by that node in the path from the root to their leaves.

### 3.2 Variant 2 – Proxy

The second variant starts from [14], an extension of Isolation Forests which improves the training stage: instead of a random train, trees are built by optimizing a predefined function. In particular, while building a tree, the authors of [14] find a split by minimizing the so-called proxy function, a function which

indicates the loss of information obtained when doing a particular split. In the classification setting, this function is typically defined using the class labels (an example is the Gini impurity): however, in the case of Isolation Forests, labels are not available and thus such function must be defined in an alternative way.

The definition given by [14] is based on the following intuition: the best split is the one which best separates instances, i.e. where one of the two children contains the maximum number of objects in a minimum volume and the other child the minimum number of instances in a maximum volume. In principle, the former child should characterize the inliers, whereas the latter should characterize the outliers. In practice, the proxy is an adaptation of the Gini impurity for the one-class context, and for its definition we need: a volume measure, the number of inliers and an estimation of the number of outliers. Aside from the number of inliers, which is known, we define the other two elements as follows:

- (i) The volume of a node  $k$  is computed via the Lebesgue measure  $Leb(k)$ . In the proxy, the ratio  $\lambda_k = \frac{Leb(k)}{Leb(parent(k))}$  between the volume of a node  $k$  and its parent is measured to retrieve the best split.
- (ii) In [14] the distribution of outliers within the node  $k$  to be split is assumed to be constant with respect to node  $k$ . Therefore the number of outliers  $n'_k$  is defined as  $n'_k = n_k \gamma$  where  $n_k$  is the number of inliers and  $\gamma$  some constant.

Leaving aside further mathematical processing steps (for all the details, please see [14]), the one-class proxy is defined as:

$$proxy(k) = \frac{n_{k_L} \gamma n_k \lambda_L}{n_{k_L} + \gamma n_k \lambda_L} + \frac{n_{k_R} \gamma n_k \lambda_R}{n_{k_R} + \gamma n_k \lambda_R} \tag{7}$$

where  $k_L$  and  $k_R$  are respectively the left and right child of  $k$  and  $\gamma = 1$ .

From our perspective, the one-class proxy can be used to measure the goodness of a split at a node  $k$  (the lower the proxy the better the split): actually, a high proxy means that the split does not separate well the data the node contains, i.e. the node is not very important in the isolation process. On the contrary a low proxy means that the node is split in a good way, i.e. some objects will likely to be isolated after it. Following this reasoning, we can define a new weight  $w_{tk}^P$ , given a tree  $t$  and a node  $k$  in  $t$ , as:

$$w_{tk}^P = \frac{1}{proxy_t(k)} \tag{8}$$

where  $proxy_t(k)$  is the proxy computed at node  $k$ .

### 3.3 Variant 3 – Proxy-Neighborhood

The third variant we propose combines the two previous versions, taking into account both the neighborhood and the proxy of a node.

Given a tree  $t$  and a node  $k$  in  $t$ , its weight  $w_{tk}^{PN}$  is:

$$w_{tk}^{PN} = \frac{1}{\text{proxy}_t(k)|N_{tk}|} \quad (9)$$

where  $\text{proxy}_t(k)$  and  $N_{tk}$  are respectively the proxy and the neighborhood of the node  $k$ .

## 4 Experimental Evaluation

The evaluation is based on 12 datasets from the UCI-ML repository [10] which are benchmarks for one-class classification [14] (they were preprocessed following the specifications in [14]). Table 1 presents an overview of the datasets. We can see that the datasets cover a large range of situations: they differ in size (the smallest one has 351 samples while the biggest 567498), in the number of features (from 3 up to 164) and in the outlier percentage (from 0.03% up to 45.8%).

Isolation Forests were trained with standard parameters, as defined in [19, 20], which are: number of objects  $N = 256$  sampled without replacement, size of the forest  $T = 100$ , number of features available per tree  $F = All$  and maximum depth  $D = \log(N)$ . In addition we also performed the experiments using  $D = N - 1$  to understand whether more descriptive trees produce better results.

Following [14], we adopted a Novelty Detection framework [25], i.e. only inliers are used in the training phase<sup>2</sup>. For each experiment the dataset has been split equally, i.e. 50% of the samples, in training and testing set. Each experiment has been repeated 30 times. Finally, as often done in many one-class classification problems [14, 19, 20] as accuracy measure we adopted the area under the ROC curve (AUC).

**Table 1.** Overview of the 12 UCI datasets used for the experimental evaluation.

Datasets	Nr. of objects	Nr. of features	Outlier %
Adult	48842	6	16.10%
Anthyroid	7200	6	7.42%
Arrhythmia	452	164	45.80%
ForestCover	286048	10	0.96%
Http	567498	3	0.39%
Ionosphere	351	32	35.90%
Pendigits	10992	16	10.41%
Pima	768	8	34.90%
Shuttle	49097	9	7.15%
Smtp	95156	3	0.03%
Spambase	4601	57	39.40%
Wilt	4839	5	5.39%

<sup>2</sup> An alternative framework to adopt would be Outlier Detection [2] which uses both outliers and inliers in the training stage.

The first analysis compares the novel anomaly score with the standard unweighted version. In Table 2 we present the results obtained when using standard parametrization with depth  $\log(N)$  and  $N - 1$ . The last row is the average across all the datasets. The best result is highlighted in bold. To assess the statistical significance we computed the standard errors of the mean, which are comprised in the range  $4 * 10^{-9}$  and  $8 * 10^{-5}$ . As a first general observation, if we look at the average across the datasets, we can see that the newly defined scores outperform the standard anomaly score. More in detail, except for two datasets, Spambase and Adult, the best score is always obtained with a path-weighted variant; for Shuttle instead, it seems there is no difference in using the standard or the novel score. On the other hand, in some cases, such as for Wilt, the improvement is quite relevant (0.718 versus 0.535). Table 2 also shows that varying the depth parameter is advantageous for our proposal, while for the anomaly score the performances vary only slightly.

**Table 2.** Results for the standard parametrization setting. Anomaly stands for the standard definition of the anomaly score, Variant 1 for the neighborhood-based variant, Variant 2 for the proxy-based one and Variant 3 for the variant based on both the neighborhood and the proxy.

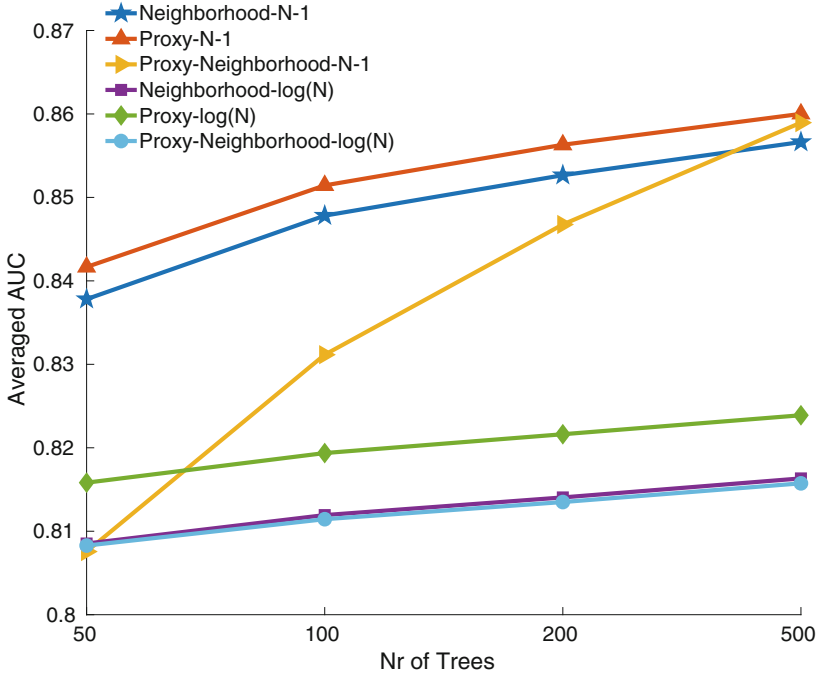
Dataset	Anomaly		Variant 1		Variant 2		Variant 3	
	$N - 1$	$\log(N)$	$N - 1$	$\log(N)$	$N - 1$	$\log(N)$	$N - 1$	$\log(N)$
Adult	<b>0.631</b>	0.630	0.625	0.629	0.627	0.630	0.610	0.629
Anthyroid	0.915	0.912	<b>0.939</b>	0.903	0.934	0.908	0.915	0.904
Arrhythmia	0.770	0.759	<b>0.773</b>	0.753	<b>0.773</b>	0.756	0.750	0.752
ForestCover	0.845	0.852	<b>0.869</b>	0.827	0.860	0.838	0.815	0.829
Http	0.994	0.993	<b>0.997</b>	0.994	<b>0.997</b>	0.996	<b>0.997</b>	0.992
Ionosphere	0.905	0.898	0.945	0.881	<b>0.959</b>	0.901	0.946	0.880
Pendigits	0.840	0.798	<b>0.928</b>	0.805	0.927	0.834	0.913	0.803
Pima	0.732	0.729	0.694	<b>0.734</b>	0.712	<b>0.734</b>	0.677	0.733
Shuttle	0.996	0.996	0.996	0.995	0.995	0.996	0.987	0.995
Smtpt	0.902	0.913	0.908	0.920	0.918	<b>0.927</b>	0.890	0.915
Spambase	0.825	<b>0.832</b>	0.798	0.827	0.824	0.829	0.755	0.828
Wilt	0.535	0.483	0.704	0.476	0.691	0.484	<b>0.718</b>	0.477
Average	0.824	0.816	0.848	0.812	<b>0.851</b>	0.819	0.831	0.811

As second experiment, we performed an analysis to study how the performances change when varying the size of the forest, i.e.  $T \in \{50, 100, 200, 500\}$ : results are presented in Table 3. For each  $T$  we report the best anomaly score (A) and the best path-weighted anomaly score (PW). We also indicate the variant reaching the best result. In bold we highlight the best result for each  $T$ . To assess the statistical significance we computed the standard errors of the



**Table 3.** Results for different  $T$ s. A stands for anomaly score, PW for path-weighted score. We report the best weighted variant between parenthesis.

Dataset	T = 50			T = 100			T = 200			T = 500		
	A	PW		A	PW		A	PW		A	PW	
Adult	<b>0.629</b>	0.628	(V2)	<b>0.631</b>	0.630	(V2)	<b>0.632</b>	0.631	(V2)	0.631	0.631	(V2)
Annthyroid	0.912	<b>0.929</b>	(V1)	0.915	<b>0.939</b>	(V1)	0.917	<b>0.943</b>	(V1)	0.918	<b>0.946</b>	(V1)
Arrhythmia	<b>0.764</b>	0.760	(V2)	0.770	<b>0.773</b>	(V2)	0.772	<b>0.778</b>	(V1)	0.775	<b>0.784</b>	(V1)
ForestCover	0.845	<b>0.849</b>	(V1)	0.852	<b>0.869</b>	(V1)	0.856	<b>0.885</b>	(V1)	0.861	<b>0.896</b>	(V1)
Http	0.992	<b>0.996</b>	(V2)	0.994	<b>0.997</b>	(V2)	0.994	<b>0.998</b>	(V2)	0.994	<b>0.998</b>	(V3)
Ionosphere	0.900	<b>0.951</b>	(V2)	0.905	<b>0.959</b>	(V2)	0.906	<b>0.961</b>	(V2)	0.907	<b>0.962</b>	(V2)
Pendigits	0.834	<b>0.913</b>	(V2)	0.840	<b>0.928</b>	(V1)	0.842	<b>0.939</b>	(V3)	0.844	<b>0.953</b>	(V3)
Pima	0.727	<b>0.730</b>	(V2)	0.732	<b>0.734</b>	(V2)	0.734	<b>0.736</b>	(V2)	0.735	<b>0.737</b>	(V2)
Shuttle	<b>0.996</b>	0.994	(V2)	0.996	0.996	(V1)	0.997	0.997	(V1)	0.997	0.997	(V1)
Smtpt	0.911	<b>0.923</b>	(V2)	0.913	<b>0.927</b>	(V2)	0.913	<b>0.927</b>	(V2)	0.913	<b>0.927</b>	(V2)
Spambase	<b>0.827</b>	0.824	(V2)	<b>0.832</b>	0.829	(V2)	<b>0.837</b>	0.834	(V2)	<b>0.840</b>	0.837	(V2)
Wilt	0.534	<b>0.687</b>	(V1)	0.535	<b>0.718</b>	(V3)	0.537	<b>0.753</b>	(V3)	0.535	<b>0.779</b>	(V3)
Average	0.823	<b>0.849</b>		0.826	<b>0.858</b>		0.828	<b>0.865</b>		0.829	<b>0.871</b>	

**Fig. 1.** Datasets average-standard parametrization

mean, which are comprised in the range  $2 * 10^{-9}$  and  $9 * 10^{-5}$ . As in Table 2, the last row presents the average across all the datasets. We can observe that the proposed method works well and the performances increase as  $T$ , the size of the forest, does. This is not true for the standard anomaly score, which improvement

reaches a plateau when  $T = 100$ . Indeed if we look at the results for  $T = 50$  the best score for 4 datasets is reached with the standard score, but if we observe the results for  $T = 500$  there is only one dataset which prefers the unweighted anomaly score. This analysis also shows that in 27/48 cases the variant based on the one-class proxy, Variant 2 is the best one. On the contrary, Variant 3 rarely achieves the best results. We can also observe that for more than half datasets the best variant does not change when varying the number of trees.

The last analysis aims at a deeper understanding of the three versions of the path-weighted anomaly score. We analysed how the performance of the different variants, averaged across all the datasets, varies when varying the size of the forest. The results are depicted in Fig. 1: we can observe that among the three variants, if we fix the depth to either  $\log(N)$  or  $N - 1$ , the best variant is in both cases the one based on the proxy, confirming the results of Table 3. It makes sense since the proxy measures the goodness of a node in terms of split, and thus how well it isolates the data. Another observation we can make is that in general, using fully grown trees, i.e. depth  $N - 1$ , increases the performances no matter which variant we consider.

## 5 Conclusions

This paper proposes an improvement of the classical anomaly score of Isolation Forests by exploiting node-related information. The proposed approach is very robust and compares well to the state of the art; in particular it achieves the best performances when working with large forests and with completely grown trees. Nevertheless we could make some further improvements, such as developing an automated method that given a dataset chooses a priori the best variant. In the future we would like to investigate novel ways to define the importance of a node and to design new methodologies to isolate points, i.e. modify the training phase.

## References

1. Bicego, M.: K-random forests: a k-means style algorithm for random forest clustering. In: International Joint Conference on Neural Networks (2019, accepted)
2. Bishop, C.M.: Novelty detection and neural network validation. IEE Proc. Vis. Image Signal Process. **141**(4), 217–222 (1994)
3. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: IEEE International Conference on Computer Vision, pp. 1–8 (2007)
4. Breiman, L., Friedman, J., Stone, C., Olshen, R.: Classification and Regression Trees. Taylor & Francis, The Wadsworth and Brooks-Cole Statistics-Probability Series (1984)
5. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
6. Chen, X., Ishwaran, H.: Random forests for genomic data analysis. Genomics **99**(6), 323–329 (2012)
7. Cl emen on, S., Depecker, M., Vayatis, N.: Ranking forests. J. Mach. Learn. Res. **14**, 39–73 (2013)

8. Désir, C., Bernard, S., Petitjean, C., Heutte, L.: One class random forests. *Pattern Recogn.* **46**, 3490–3506 (2013)
9. Ding, Z., Fei, M.: An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proc. Volumes* **46**(20), 12–17 (2013)
10. Dua, D., Graff, C.: UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
11. 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)
12. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **15**(1), 3133–3181 (2014)
13. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. *Mach. Learn.* **63**(1), 3–42 (2006)
14. Goix, N., Drougard, N., Brault, R., Chiapino, M.: One class splitting criteria for random forests. In: *ACML*, pp. 1–16 (2017)
15. Guha, S., Mishra, N., Roy, G., Schrijvers, O.: Robust random cut forest based anomaly detection on streams. In: *ICML* (2016)
16. Ishwaran, H., Kogalur, U.B., Blackstone, E.H., Lauer, M.S., et al.: Random survival forests. *Ann. Appl. Stat.* **2**(3), 841–860 (2008)
17. Joly, A., Geurts, P., Wehenkel, L.: Random forests with random projections of the output space for high dimensional multi-label classification. In: *Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI)*, vol. 8724, pp. 607–622. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-662-44848-9\\_39](https://doi.org/10.1007/978-3-662-44848-9_39)
18. Kittler, J.: Combining classifiers: a theoretical framework. *Pattern Anal. Appl.* **1**(1), 18–27 (1998)
19. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: *IEEE International Conference on Data Mining*, IEEE Computer Society, pp. 413–422 (2008)
20. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data* **6**(1), 3:1–3:39 (2012)
21. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: *Advances in Neural Information Processing Systems*, pp. 985–992 (2007)
22. Moya, M.M., Koch, M.W., Hostetler, L.D.: One-class classifier networks for target recognition applications. *NASA STI/Recon Technical Report N* **93** (1993)
23. Pal, M.: Random forest classifier for remote sensing classification. *Int. J. Remote Sens.* **26**(1), 217–222 (2005)
24. Preiss, B.R.: *Data Structures and Algorithms*. Wiley (1999)
25. Ritter, G., Gallegos, M.T.: Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Pattern Recogn. Lett.* **18**(6), 525–539 (1997)
26. Shi, T., Horvath, S.: Unsupervised learning with random forest predictors. *J. Comput. Graph. Stat.* **15**, 118–138 (2005)
27. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2008)
28. Susto, G.A., Beghi, A., McLoone, S.: Anomaly detection through on-line isolation forest: an application to plasma etching. In: *Annual SEMI Advanced Semiconductor Manufacturing Conference* (2017)

29. Tax, D.: One-class classification; concept-learning in the absence of counter-examples. Ph.D. thesis, Delft University of Technology (2001)
30. Zhu, X., Loy, C.C., Gong, S.: Constructing robust affinity graphs for spectral clustering. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1450–1457 (2014)