One Class Splitting Criteria for Random Forests

Nicolas Goix NICOLAS.GOIX@TELECOM-PARISTECH.FR
LTCI, Télécom ParisTech, Université Paris-Saclay, France.

Nicolas Drougard NICOLAS.DROUGARD@ISAE-SUPAERO.FR
ISAE-Supaero, Toulouse, France

Romain Brault ROMAIN.BRAULT@TELECOM-PARISTECH.FR
LTCI, Télécom ParisTech, Université Paris-Saclay, France.

Mael Chiapino MAEL.CHIAPINO@TELECOM-PARISTECH.FR
LTCI, Télécom ParisTech, Université Paris-Saclay, France.

Editors: Yung-Kyun Noh and Min-Ling Zhang

Abstract

Random Forests (RFs) are strong machine learning tools for classification and regression. However, they remain supervised algorithms, and no extension of RFs to the one-class setting has been proposed, except for techniques based on second-class sampling. This work fills this gap by proposing a natural methodology to extend standard splitting criteria to the one-class setting, structurally generalizing RFs to one-class classification. An extensive benchmark of seven state-of-the-art anomaly detection algorithms is also presented. This empirically demonstrates the relevance of our approach.

Keywords: Random Forests, One-Class Classification, Anomaly Detection

1. Introduction

Anomalies, novelties or outliers are usually assumed to lie in low probability regions of the data generating process. This assumption drives many statistical anomaly detection methods. Parametric techniques Barnett and Lewis (1994); Eskin (2000) suppose that the inliers are generated by a distribution belonging to some specific parametric model a priori known. Here and hereafter, we denote by inliers the 'not abnormal' data, and by outliers/anomalies/novelties the data from the abnormal class. Classical non-parametric approaches are based on density (level set) estimation Schölkopf et al. (2001); Scott and Nowak (2006); Breunig et al. (2000); Quinn and Sugiyama (2014), on dimensionality reduction Shyu et al. (2003); Aggarwal and Yu (2001) or on decision trees Liu et al. (2008); Shi and Horvath (2012). Relevant overviews of current research on anomaly detection can be found in Hodge and Austin (2004); Chandola et al. (2009); Patcha and Park (2007); Markou and Singh (2003).

The algorithm proposed in this paper lies in the *novelty detection* setting, also called *one-class classification*. In this framework, we assume that we only observe examples of one class (referred to as the normal class, or inlier class). The second (hidden) class is called the abnormal class, or outlier class. The goal is to identify characteristics of the inlier class, such as its support or some density level sets with levels close to zero. This setup is for instance used in some (non-parametric) kernel methods such as One-Class Support Vector Machine

(OCSVM) Schölkopf et al. (2001), which extends the SVM methodology Cortes and Vapnik (1995); Shawe-Taylor and Cristianini (2004) to handle training using only inliers. More recently, Least Squares Anomaly Detection (LSAD) Quinn and Sugiyama (2014) similarly extends a multi-class probabilistic classifier Sugiyama (2010) to the one-class setting.

Random Forests (RFs) are strong machine learning tools Breiman (2001), comparing well with state-of-the-art methods such as SVM or boosting algorithms Freund et al. (1996), and used in a wide range of domains Svetnik et al. (2003); Díaz-Uriarte and De Andres (2006); Genuer et al. (2010). These estimators fit a number of decision tree classifiers on different random sub-samples of the dataset. Each tree is built recursively, according to a splitting criterion based on some impurity measure of a node. The prediction is done by an average over each tree prediction. In classification the averaging is based on a majority vote. Practical and theoretical insights on RFs are given in Genuer et al. (2008); Biau et al. (2008); Louppe (2014); Biau and Scornet (2016).

Yet few attempts have been made to transfer the idea of RFs to one-class classification Liu et al. (2008); Shi and Horvath (2012); Désir et al. (2013); Guha and Schrijvers (2016). In Liu et al. (2008), the novel concept of *isolation* is introduced: the Isolation Forest algorithm isolates anomalies, instead of profiling the inlier behavior which is the usual approach. It avoids adapting splitting rules to the one-class setting by using extremely randomized trees, also named extra trees Geurts et al. (2006): isolation trees are built completely randomly, without any splitting rule. Therefore, Isolation Forest is not really based on RFs, the base estimators being extra trees instead of classical decision trees. Isolation Forest performs very well in practice with low memory and time complexities. RecentlyGuha and Schrijvers (2016) proposed an extension of Isolation Forest to streaming data. Weights on dimensions are added to chose (still at random) the dimension to split on.

In Désir et al. (2013); Shi and Horvath (2012), outliers are generated to artificially form a second class. In Désir et al. (2013) the authors propose a technique to reduce the number of outliers needed by shrinking the dimension of the input space. The outliers are then generated from the reduced space using a distribution complementary to the inlier distribution. Thus their algorithm artificially generates a second class, in order to use classical RFs. In Shi and Horvath (2012), two different outliers generating processes are compared. In the first one, an artificial second class is created by randomly sampling from the product of empirical marginal (inlier) distributions. In the second one outliers are uniformly generated from the hyper-rectangle that contains the observed data. The first option is claimed to work best in practice, which can be understood from the curse of dimensionality argument: in large dimension Tax and Duin (2002), when the outliers distribution is not tightly defined around the target set, the chance for an outlier to be in the target set becomes very small, so that a huge number of outliers has to be sampled.

Looking beyond the RF literature, Scott and Nowak (2006) proposes a methodology to build dyadic decision trees to estimate minimum-volume sets Polonik (1997); Einmahl and Mason (1992). This is done by reformulating their structural risk minimization problem to be able to use the algorithm in Blanchard et al. (2004). While this methodology can also be used for non-dyadic trees pruning (assuming such a tree has been previously constructed, *e.g.* using some greedy heuristic), it does not allow to grow such trees. Also, the theoretical guaranties derived there relies on the dyadic structure assumption. In the same spirit, Clémençon and Robbiano (2014) proposes to use the two-class splitting criterion defined in Clémençon and Vayatis (2009). This two-class splitting rule aims at producing oriented decision trees with a 'left-to-right' structure to address the bipartite ranking task. Extension to the one-class setting is done by assuming a uniform distribution for the outlier class. Consistency and rate bounds relies also on this left-to-right structure. Thus, these two references Scott and Nowak (2006); Clémençon and Robbiano (2014) impose constraints on the tree structure (designed to allow a statistical study) which differs then significantly from the general structure of the base estimators in RF. The price to pay is a lack of flexibility of the model affecting its ability to capture complex broader patterns or structural characteristics from the data.

In this paper, we make the choice to stick to the RF framework. We do not assume any structure for the binary decision trees. The price to pay is a lack of statistical guaranties – the consistency of RFs has only been proved in the context of regression additive models Scornet et al. (2015). The gain is that we preserve the flexibility and strength of RFs, the algorithm presented here being able to compete well with state-of-the-art anomaly detection algorithms. Besides, we do not assume any (fixed in advance) outlier distribution as in Clémençon and Robbiano (2014), but define it in an adaptive way during the tree building process.

To the best of our knowledge, no algorithm structurally extends (without second class sampling and without alternative base estimators) RFs to one-class classification. Here we precisely introduce such a methodology. It builds on a natural adaptation of two-class splitting criteria to the one-class setting, as well as an adaptation of the two-class majority vote.

Basic idea. To split a node without second class examples (outliers), we proceed as follows. Each time we look for the best split for a node t, we simply replace (in the two-class *impurity decrease* to be maximized) the second class proportion going to the left child node t_L by the proportion expectation $\mathrm{Leb}(t_L)/\mathrm{Leb}(t)$ (idem for the right node), $\mathrm{Leb}(t)$ being the volume of the rectangular cell corresponding to node t. It ensures that one child node manages to capture the maximum number of observations with a minimal volume, while the other child looks for the opposite.

This simple idea corresponds to an adaptive modeling of the outlier distribution. The proportion expectation mentioned above is weighted proportionally to the number of inliers in node t. Thus, the resulting outlier distribution is tightly concentrated around the inliers. This resulting outlier distribution can be seen as a 'worse case' / 'most difficult' distribution, used to learn accurately the inlier distribution. Besides, and this attests the consistency of our approach with the two-class framework, it turns out that the one-class model promoted here corresponds to the asymptotic behavior of an adaptive outliers generating methodology.

This paper is structured as follows. Section 2 provides the reader with necessary background, to address Section 3 which proposes an adaptation of RFs to the one-class setting and describes a generic one-class random forest algorithm. The latter is compared empirically with state-of-the-art anomaly detection methods in Section 4. Finally a theoretical justification of the one-class criterion is given in Section 5.

2. Background on decision trees

Let us denote by $\mathcal{X} \subset \mathbb{R}^d$ the d-dimensional hyper-rectangle containing all the observations. Consider a binary tree on $\mathcal{X}$ whose node values are subsets of $\mathcal{X}$, iteratively produced by splitting $\mathcal{X}$ into two disjoint subsets. Each internal node t with value $\mathcal{X}_t$ is labeled with a split feature m_t and split value c_t (along that feature), in such a way that it divides $\mathcal{X}_t$ into two disjoint spaces $\mathcal{X}_{t_L} := \{x \in \mathcal{X}_t, x_{m_t} < c_t\}$ and $\mathcal{X}_{t_R} := \{x \in \mathcal{X}_t, x_{m_t} \geq c_t\}$, where t_L (resp. t_R) denotes the left (resp. right) children of node t, and x_j denotes the jth coordinate of vector x. Such a binary tree is grown from a sample $X_1, \ldots, X_n$ ($\forall i$, $X_i \in \mathcal{X}$) and its finite depth is determined either by a fixed maximum depth value or by a stopping criterion evaluated on the nodes (*e.g.* based on an impurity measure). The external nodes (the *leaves*) form a partition of $\mathcal{X}$.

In a supervised classification setting, these binary trees are called *classification trees* and prediction is made by assigning to new observation $x \in \mathcal{X}$ the majority class of the leaves containing x. This is called the *majority vote*. Classification trees are usually built using an impurity measure $i(t)$ whose decrease is maximized at each split of a node t, yielding an optimal split (m_t^*, c_t^*). The decrease of impurity (also called *goodness of split*) $\Delta i(t, t_L, t_R)$ *w.r.t.* the split (m_t, c_t) and corresponding to the partition $\mathcal{X}_t = \mathcal{X}_{t_L} \sqcup \mathcal{X}_{t_R}$ of the node t is defined as

$$\Delta i(t, t_L, t_R) = i(t) - p_L i(t_L) - p_R i(t_R), \tag{1}$$

where $p_L = p_L(t)$ (resp. $p_R = p_R(t)$) is the proportion of samples from $\mathcal{X}_t$ going to $\mathcal{X}_{t_L}$ (resp. to $\mathcal{X}_{t_R}$). The impurity measure $i(t)$ reflects the goodness of node t: the smaller $i(t)$, the purer the node t and the better the prediction by majority vote on this node. Usual choices for $i(t)$ are the Gini index Gini (1912) or the Shannon entropy Shannon (2001). To produce a randomized tree, these optimization steps are usually partially randomized (conditionally on the data, splits (m_t^*, c_t^*)'s become random variables). A classification tree can even be grown totally randomly Geurts et al. (2006). In a two-class classification setup, the Gini index is

$$i_G(t) = 2\left(\frac{n_t}{n_t + n_t'}\right)\left(\frac{n_t'}{n_t + n_t'}\right) \tag{2}$$

where n_t (resp. n_t') stands for the number of observations with label 0 (resp. 1) in node t. The Gini index is maximal when $n_t/(n_t + n_t') = n_t'/(n_t + n_t') = 0.5$, namely when the conditional probability to have label 0 given that we are in node t is the same as to have label 0 unconditionally: the node t does not discriminate at all between the two classes.

For a node t, maximizing the impurity decrease (1) is equivalent to minimizing $p_L i(t_L) + p_R i(t_R)$. Since $p_L = (n_{t_L} + n'_{t_L})/(n_t + n_t')$ and $p_R = (n_{t_R} + n'_{t_R})/(n_t + n_t')$, and the quantity $(n_t + n_t')$ being constant in the optimization problem, this is equivalent to minimizing the following proxy of the impurity decrease,

$$I(t_L, t_R) = (n_{t_L} + n'_{t_L}) i(t_L) + (n_{t_R} + n'_{t_R}) i(t_R). \tag{3}$$

Note that with the Gini index $i_G(t)$ given in (2), the corresponding proxy of the impurity decrease is

$$I_G(t_L, t_R) = \frac{n_{t_L} n'_{t_L}}{n_{t_L} + n'_{t_L}} + \frac{n_{t_R} n'_{t_R}}{n_{t_R} + n'_{t_R}}. \tag{4}$$

In the one-class setting, no label is available, hence the impurity measure $i(t)$ does not apply to this setup. The standard splitting criterion which consists in minimizing the latter cannot be used anymore.

3. Adaptation to the one-class setting

The two reasons why RFs do not apply to one-class classification are that the standard splitting criterion does not apply to this setup, as well as the majority vote. In this section, we propose a one-class splitting criterion and a one-class version of the majority vote.

3.1. One-class splitting criterion

As one does not observe the second-class (outliers), n_t' needs to be defined. In the naive approach below, it is defined as $n_t' := n'\mathrm{Leb}(\mathcal{X}_t)/\mathrm{Leb}(\mathcal{X})$, where n' is the assumed total number of (hidden) outliers. Here and hereafter, Leb denotes the Lebesgue measure on $\mathbb{R}^d$. In the adaptive approach hereafter, it is defined as $n_t' := \gamma n_t$, with typically $\gamma = 1$. Thus, the class ratio $\gamma_t := n_t'/n_t$ is well defined in both approaches and goes to 0 when $\mathrm{Leb}(\mathcal{X}_t) \to 0$ in the naive approach, while it is maintained constant to γ in the adaptive one.

Naive approach. A naive approach to extend the Gini splitting criterion to the one-class setting is to assume a uniform distribution for the second class (outliers), and to replace their number n_t' in node t by the expectation $n'\mathrm{Leb}(\mathcal{X}_t)/\mathrm{Leb}(\mathcal{X})$, where n' denotes the total number of outliers (for instance, it can be chosen as a proportion of the number of inliers). The problem with this approach appears when the dimension is *not small*. As mentioned in the introduction (curse of dimensionality), when actually generating n' uniform outliers on $\mathcal{X}$, the probability that a node (sufficiently small to yield a good precision) contains at least one of them is very close to zero. That is why data-dependent distributions for the outlier class are often considered Désir et al. (2013); Shi and Horvath (2012). Taking the expectation $n'\mathrm{Leb}(\mathcal{X}_t)/\mathrm{Leb}(\mathcal{X})$ to replace the number of points in node t does not solve the curse of dimensionality mentioned in the introduction: the volume proportion $L_t := \mathrm{Leb}(\mathcal{X}_t)/\mathrm{Leb}(\mathcal{X})$ is very close to 0 for nodes t deep in the tree, especially in large dimension. In addition, we typically grow trees on sub-samples of the input data, meaning that even the root node of the trees may be very small compared to the hyper-rectangle containing all the input data. An other problem is that the Gini splitting criterion is skew-sensitive Flach (2003), and has here to be applied on nodes t with $0 \simeq n_t' \ll n_t$. When trying empirically this approach, we observe that splitting such nodes produces a child containing (almost) all the data (see Section 5).

Example 1 *To illustrate the fact that the volume proportion $L_t := Leb(\mathcal{X}_t)/Leb(\mathcal{X})$ becomes very close to zero in large dimension for lots of nodes t (in particular the leaves), suppose for the sake of simplicity that the input space is $\mathcal{X} = [0,1]^d$. Suppose that we are looking for a rough precision of $1/2^3 = 0.125$ in each dimension,* i.e. *a unit cube precision of 2^{-3d}. To achieve such a precision, the splitting criterion has to be used on nodes/cells t of volume of order 2^{-3d}, namely with $L_t = 1/2^{3d}$. Note that if we decide to choose n' to be 2^{3d} times larger than the number of inliers in order that $n'L_t$ is not negligible* w.r.t. *the number* *of inliers, the same (reversed) problem of unbalanced classes appears on nodes with small depth.*

Adaptive approach. Our solution is to remove the uniform assumption on the outliers, and to choose their distribution adaptively in such a way it is tightly concentrated around the inlier distribution. Formally, the idea is to maintain constant the class ratio $\gamma_t := n'_t/n_t$ on each node t: before looking for the best split, we update the number of outliers to be equal (up to a scaling constant γ) to the number of inliers, $n'_t = \gamma n_t$, *i.e.* $\gamma_t \equiv \gamma$. These (hidden) outliers are uniformly distributed on node t. The parameter γ is typically set to $\gamma = 1$, see supplementary material Section A.1 for a discussion on the relevance of this choice (in a nutshell, γ has an influence on optimal splits).

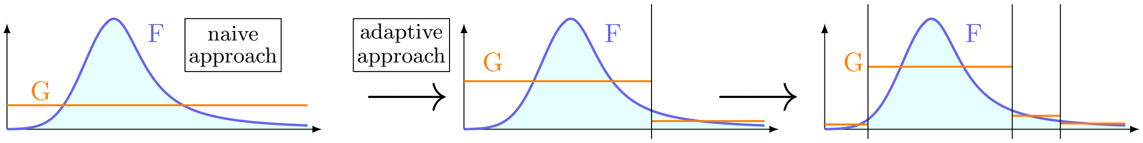

Figure 1: Outliers distribution G in the naive and adaptive approach. In the naive approach, G does not depend on the tree and is constant on the input space. In the adaptive approach the distribution depends on the inlier distribution F through the tree. The outliers density is constant and equal to the average of F on each node before splitting it.

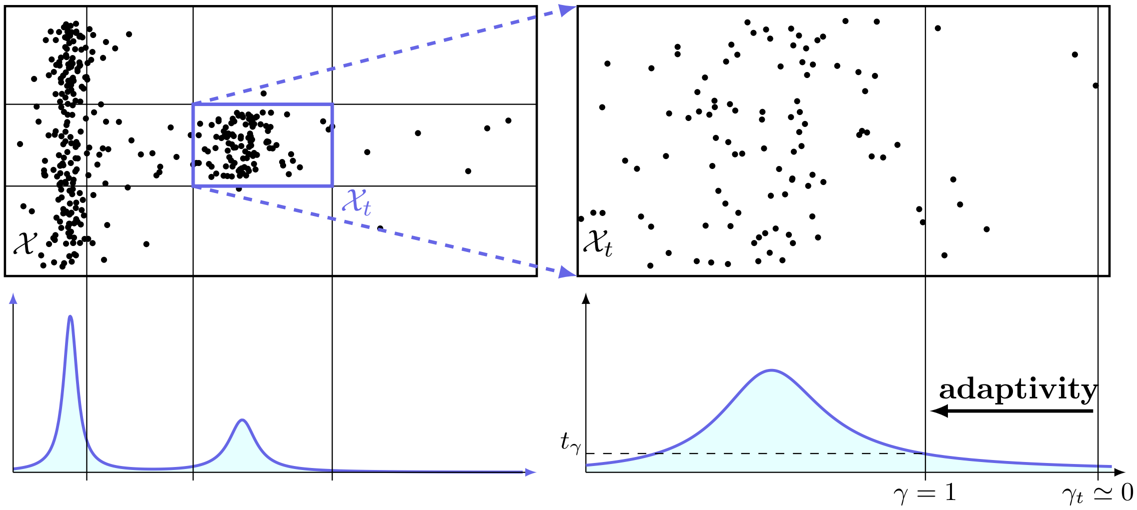

Figure 2: The left part represents the dataset under study and the underlying density. The node $\mathcal{X}_t$ obtained after some splits is illustrated in the right part of this figure: without the proposed adaptive approach, the class ratio γ_t becomes too small and yields poor splits (all the data are in the 'inlier side' of the split, which thus does not discriminate at all). Contrariwise, setting γ to one, *i.e.* using the adaptive approach, is far preferable.

With this methodology, one cannot derive a one-class version of the Gini index (2), but we can define a one-class version of the proxy of the impurity decrease (4), by simply replacing n'_{t_L} (resp. n'_{t_R}) by $n'_t\lambda_L$ (resp. $n'_t\lambda_R$), where $\lambda_L := \mathrm{Leb}(\mathcal{X}_{t_L})/\mathrm{Leb}(\mathcal{X}_t)$ and $\lambda_R := \mathrm{Leb}(\mathcal{X}_{t_R})/\mathrm{Leb}(\mathcal{X}_t)$ are the volume proportion of the two child nodes:

$$I_G^{OC-ad}(t_L, t_R) = \frac{n_{t_L}\gamma n_t \lambda_L}{n_{t_L} + \gamma n_t \lambda_L} + \frac{n_{t_R}\gamma n_t \lambda_R}{n_{t_R} + \gamma n_t \lambda_R}. \tag{5}$$

Minimization of the one-class Gini improvement proxy (5) is illustrated in Figure 2. Note that $n'_t\lambda_L$ (resp. $n'_t\lambda_R$) is the expectation of the number of uniform observations (on $\mathcal{X}_t$) among n'_t (fixed to $n'_t = \gamma n_t$) falling into the left (resp. right) node.

Choosing the split minimizing $I_G^{OC-ad}(t_L, t_R)$ at each step of the tree building process, corresponds to generating $n'_t = \gamma n_t$ outliers each time the best split has to be chosen for node t, and then using the classical two-class Gini proxy (4). The only difference is that n'_{t_L} and n'_{t_R} are replaced by their expectations $n'_t\lambda_{t_L}$ and $n'_t\lambda_{t_R}$ in our method.

Resulting outlier distribution. Figure 1 shows the corresponding outlier density G (we drop the dependence in the number of splits to keep the notations uncluttered). Note that G is a piece-wise constant approximation of the inlier distribution F. Considering the Neyman-Pearson test $X \sim F$ vs. $X \sim G$ instead of $X \sim F$ vs. $X \sim \mathrm{Unif}$ may seem surprising at first sight. Let us try to give some intuition on why this works in practice. First, there exists (at each step) $\epsilon > 0$ such that $G > \epsilon$ on the entire input space, since the density G is constant on each node and equal to the average of F on this node *before splitting it*. If the average of F was estimated to be zero (no inlier in the node), the node would obviously not have been splitted, from where the existence of ϵ. Thus, at each step, one can also view G as a piece-wise approximation of $F_\epsilon := (1-\epsilon)F + \epsilon\mathrm{Unif}$, which is a mixture of F and the uniform distribution. Yet, one can easily show that optimal tests for the Neyman-Pearson problem $H_0 : X \sim F$ vs. $H_1 : X \sim F_\epsilon$ are identical to the optimal tests for $H_0 : X \sim F$ vs. $H_1 : X \sim \mathrm{Unif}$, since the corresponding likelihood ratios are related by a monotone transformation, see Scott and Blanchard (2009) for instance (in fact, this reference shows that these two problems are even equivalent in terms of consistency and rates of convergence of the learning rules). An other intuitive justification is as follows. In the first step, the algorithm tries to discriminate F from Unif. When going deeper in the tree, splits manage to discriminate F from a (more and more accurate) approximation of F. Asymptotically, splits become unrelevant since they are trying to discriminate F from itself (a perfect approximation, $\epsilon \to 0$).

Remark 1 *(*Consistency with the two-class framework*) Consider the following method to generate outliers – tightly concentrated around the support of the inlier distribution. Sample uniformly $n' = \gamma n$ outliers on the rectangular cell containing all the inliers. Split this root node using classical two-class impurity criterion (*e.g.* minimizing (4)). Apply recursively the three following steps: for each node t, remove the potential outliers inside $\mathcal{X}_t$, re-sample $n'_t = \gamma n_t$ uniform outliers on $\mathcal{X}_t$, and use the latter to find the best split using (4). Then, each optimization problem (4) we have solved is equivalent (in expectation) to its one-class version (5). In other words, by generating outliers adaptively, we can recover (in average) a tree grown using the one-class impurity, from a tree grown using the two-class impurity.*

Remark 2 *(*Extension to other impurity criteria*) Our extension to the one-class setting also applies to other impurity criteria. For instance, in the case of the Shan-*

non entropy defined in the two-class setup by $i_S(t) = \frac{n_t}{n_t+n_t'} \log_2 \frac{n_t+n_t'}{n_t} + \frac{n_t'}{n_t+n_t'} \log_2 \frac{n_t+n_t'}{n_t'}$, *the one-class impurity improvement proxy becomes* $I_S^{OC-ad}(t_L, t_R) = n_{t_L} \log_2 \frac{n_{t_L}+\gamma n_t \lambda_L}{n_{t_L}} + n_{t_R} \log_2 \frac{n_{t_R}+\gamma n_t \lambda_R}{n_{t_R}}$.

3.2. Prediction: scoring function of the forest

Now that RFs can be grown in the one-class setting using the one-class splitting criterion, the forest has to return a prediction adapted to this framework. In other words we also need to extend the concept of majority vote. Most usual one-class (or more generally anomaly detection) algorithms actually provide more than just a level-set estimate or a predicted label for any new observation, abnormal vs. normal. Instead, they return a real valued function, termed *scoring function*, defining a pre-order/ranking on the input space. Such a function $s : \mathbb{R}^d \to \mathbb{R}$ allows to rank any observations according to their supposed 'degree of abnormality'. Thresholding it provides level-set estimates, as well as a decision rule that splits the input space into inlier/normal and outlier/abnormal regions. The scoring function $s(x)$ we use is the one defined in Liu et al. (2008) in view of its established high performance. It is a decreasing function of the average depth of the leaves containing x in the forest. An average term is added to each node containing more than one sample, say containing N samples. This term $c(N)$ is the average depth of an extremely randomized tree Geurts et al. (2006) (*i.e.* built without minimizing any criterion, by randomly choosing one feature and one uniform value over this feature to split on) on N samples. Formally,

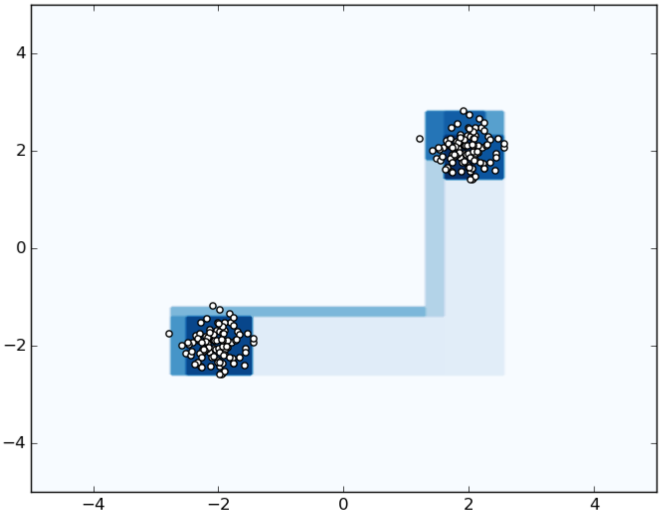

Figure 3: OneClassRF with one tree: level-sets of the scoring function.

$$\log_2 s(x) = -\left(\sum_{t \text{ leaves}} \mathbb{1}_{\{x \in t\}} d_t + c(n_t) \right) \ / \ c(n), \tag{6}$$

where d_t is the depth of node t, and $c(n) = 2H(n-1) - 2(n-1)/n$, $H(i)$ being the harmonic number. Alternative scoring functions can be defined for this one-class setting (cf. supplementary material Section A.2).

3.3. OneClassRF: a Generic One-Class Random Forest algorithm

Let us summarize the One Class Random Forest (OneClassRF) algorithm, based on generic RFs Breiman (2001). It has 6 parameters: *max_samples*, *max_features_tree*, *max_features_node*, γ, *max_depth*, *n_trees*. Each tree is classically grown on a random subset of both the input samples and the input features Ho (1998); Panov and Džeroski (2007). This random subset is a sub-sample of size *max_samples*, with *max_features_tree* variables chosen at random without replacement (replacement is only done after the tree is grown). The tree is built by minimizing (5) for each split, using parameter γ (recall that $n_t' := \gamma n_t$), until either the maximal depth *max_depth* is achieved or the node contains only one point. Minimizing (5) is done as introduced in Amit and Geman (1997): at each node, we search the best split over a random selection of features with fixed size *max_features_node*. The forest is composed of a number *n_trees* of trees. The predicted score of a point x is given by $s(x)$, with s defined by (6). Remarks on alternative stopping criteria and variable importances are available in supplementary material Section A.3.

Figure 3 represents the level sets of the scoring function produced by OneClassRF, with only one tree of maximal depth 4, without sub-sampling, and using the Gini-based one-class splitting criterion with $\gamma = 1$.

4. Benchmark

In this section, we compare the OneClassRF algorithm described above to seven state-of-art anomaly detection algorithms [1]: the isolation forest algorithm Liu et al. (2008) (iForest), a one-class RFs algorithm based on sampling a second-class Désir et al. (2013) (OCRFsampling), one class SVM Schölkopf et al. (2001) (OCSVM), local outlier factor Breunig et al. (2000) (LOF), Orca Bay and Schwabacher (2003), Least Squares Anomaly Detection Quinn and Sugiyama (2014) (LSAD), Random Forest Clustering Shi and Horvath (2012) (RFC).

4.1. Default parameters of OneClassRF

The default parameters taken for our algorithm are the followings. *max_samples* is fixed to 20% of the training sample size (with a minimum of 100); *max_features_tree* is fixed to 50% of the total number of features with a minimum of 5 (*i.e.* each tree is built on 50% of the total number of features); *max_features_node* is fixed to 5; γ is fixed to 1; *max_depth* is fixed to $\log_2$ (logarithm in base 2) of the training sample size as in Liu et al. (2008); *n_trees* is fixed to 100 as in the previous reference.

The other algorithms in the benchmark are trained with their recommended (default) hyper-parameters as seen in their respective paper or author's implementation. See supplementary material Section B for details. The characteristics of the twelve reference datasets considered here are summarized in Table 1. They are all available on the UCI repository

[1] For the sake of reproducibility, all the code used to obtain the numerical results of this section is available at: `https://github.com/ngoix/OCRF`

Table 1: Original datasets characteristics

Datasets	nb of samples	nb of features	anomaly class	
adult	48842	6	class '$> 50K$'	(23.9%)
annthyroid	7200	6	classes $\neq 3$	(7.42%)
arrhythmia	452	164	classes $\neq 1$ (features 10-14 removed)	(45.8%)
forestcover	286048	10	class 4 (vs. class 2)	(0.96%)
http	567498	3	attack	(0.39%)
ionosphere	351	32	bad	(35.9%)
pendigits	10992	16	class 4	(10.4%)
pima	768	8	pos (class 1)	(34.9%)
shuttle	85849	9	classes $\neq 1$ (class 4 removed)	(7.17%)
smtp	95156	3	attack	(0.03%)
spambase	4601	57	spam	(39.4%)
wilt	4839	5	class 'w' (diseased trees)	(5.39%)

Lichman (2013) and the preprocessing is done as usually in the litterature (see supplementary material Section C).

Table 2: Results for the novelty detection setting (novelty detection framework).

Datasets	OneClassRF		iForest		OCRFsampl.		OCSVM		LOF		Orca		LSAD		RFC	
	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR	ROC	PR
adult	**0.665**	**0.278**	0.661	0.227	NA	NA	0.638	0.201	0.615	0.188	0.606	0.218	0.647	0.258	NA	NA
annthyroid	**0.936**	0.468	0.913	0.456	0.918	**0.532**	0.706	0.242	0.832	0.446	0.587	0.181	0.810	0.327	NA	NA
arrhythmia	0.684	0.510	0.763	0.492	0.639	0.249	**0.922**	**0.639**	0.761	0.473	0.720	0.466	0.778	0.514	0.716	0.299
forestcover	0.968	0.457	0.863	0.046	NA	NA	NA	NA	**0.990**	**0.795**	0.946	0.558	0.952	0.166	NA	NA
http	**0.999**	**0.838**	0.994	0.197	NA	NA	NA	NA	NA	NA	**0.999**	0.812	0.981	0.537	NA	NA
ionosphere	0.909	0.643	0.902	0.535	0.859	0.609	0.973	0.849	0.959	0.807	0.928	**0.910**	**0.978**	0.893	0.950	0.754
pendigits	0.960	0.559	0.810	0.197	0.968	0.694	0.603	0.110	0.983	0.827	**0.993**	**0.925**	0.983	0.752	NA	NA
pima	0.719	0.247	0.726	0.183	**0.759**	**0.266**	0.716	0.237	0.700	0.152	0.588	0.175	0.713	0.216	0.506	0.090
shuttle	**0.999**	**0.998**	0.996	0.973	NA	NA	0.992	0.924	**0.999**	0.995	0.890	0.782	0.996	0.956	NA	NA
smtp	0.922	0.499	0.907	0.005	NA	NA	0.881	**0.656**	**0.924**	0.149	0.782	0.142	0.877	0.381	NA	NA
spambase	**0.850**	0.373	0.824	0.372	0.797	**0.485**	0.737	0.208	0.746	0.160	0.631	0.252	0.806	0.330	0.723	0.151
wilt	0.593	0.070	0.491	0.045	0.442	0.038	0.323	0.036	0.697	0.092	0.441	0.030	0.677	0.074	**0.896**	**0.631**
average:	**0.850**	**0.495**	0.821	0.311	0.769	0.410	0.749	0.410	0.837	0.462	0.759	0.454	**0.850**	0.450	0.758	0.385
cum. train time:	**61s**		68s		NA		NA		NA		2232s		73s		NA	

4.2. Results

The experiments are performed in the novelty detection framework, where the training set consists of inliers only. For each algorithm, 10 experiments on random training and testing datasets are performed, yielding averaged ROC and Precision-Recall curves whose AUCs are summarized in Table 2 (higher is better). The training time of each algorithm has been limited (for each experiment among the 10 performed for each dataset) to 30 minutes, where 'NA' indicates that the algorithm could not finish training within the allowed time limit. In average on all the datasets, our proposed algorithm 'OneClassRF' achieves both best AUC ROC and AUC PR scores (with LSAD for AUC ROC). It also achieves the lowest cumulative training time. For further insights on the benchmarks cf. supplementary material Section A.

It appears that OneClassRF has the best performance on five datasets in terms of ROC AUCs, and is also the best in average. Computation times (training plus testing) of OneClassRF are also very competitive. Experiments in an outlier detection framework (the training set is polluted by outliers) have also been made (see supplementary material Section D). The anomaly rate is arbitrarily bounded to 10% max (before splitting data into training and testing sets).

5. Theoretical analysis

This section aims at recovering (5) from a natural modeling of the one-class framework, along with a theoretical study of the problem raised by the naive approach.

5.1. Underlying model

In order to generalize the two-class framework to the one-class one, we need to consider the population versions associated to empirical quantities (1), (2) and (3), as well as the underlying model assumption. The latter can be described as follows.

Existing Two-Class Model (n, $\boldsymbol{\alpha}$). We consider a *r.v.* $X : \Omega \to \mathbb{R}^d$ *w.r.t.* a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The law of X depends on another *r.v.* $y \in \{0,1\}$, verifying $\mathbb{P}(y=1) = 1 - \mathbb{P}(y=0) = \alpha$. We assume that conditionally on $y=0$, X follows a law F, and conditionally on $y=1$ a law G;

$$\begin{array}{lcll} X \mid y=0 & \sim & F, & \qquad \mathbb{P}(y=0) = 1-\alpha, \\ X \mid y=1 & \sim & G, & \qquad \mathbb{P}(y=1) = \alpha. \end{array}$$

Then, considering $p(t_L|t) = \mathbb{P}(X \in \mathcal{X}_{t_L} | X \in \mathcal{X}_t)$, $p(t_R|t) = \mathbb{P}(X \in \mathcal{X}_{t_R} | X \in \mathcal{X}_t)$, the population version (probabilistic version) of (1) is

$$\Delta i^{theo}(t, t_L, t_R) \;=\; i^{theo}(t) \;-\; p(t_L|t) i^{theo}(t_L) - \; p(t_R|t) i^{theo}(t_R). \tag{7}$$

It can be used with the Gini index i_G^{theo},

$$i_G^{theo}(t) \;=\; 2\mathbb{P}(y=0|X \in \mathcal{X}_t) \cdot \mathbb{P}(y=1|X \in \mathcal{X}_t) \tag{8}$$

which is the population version of (2).

One-Class-Model ($\boldsymbol{n}$, $\boldsymbol{\alpha}$). We model the one-class framework as follows. Among the n *i.i.d.* observations, we only observe those with $y=0$ (the inliers), namely N realizations of $(X \mid y=0)$, where N is itself a realization of a *r.v.* $\mathbf{N}$ of law $\mathbf{N} \sim \text{Bin}\big(n, (1-\alpha)\big)$. Here and hereafter, $\text{Bin}(n,p)$ denotes the binomial distribution with parameters (n,p). As outliers are not observed, it is natural to assume that G follows a uniform distribution on the hyper-rectangle $\mathcal{X}$ containing all the observations, so that G has a constant density $g(x) \equiv 1/\text{Leb}(\mathcal{X})$ on $\mathcal{X}$. Note that this assumption *will be removed* in the adaptive approach described below – which aims at maintaining a non-negligible proportion of (hidden) outliers in every nodes.

Let us define $L_t = \text{Leb}(\mathcal{X}_t)/\text{Leb}(\mathcal{X})$. Then, $\mathbb{P}(X \in \mathcal{X}_t,\; y=1) = \mathbb{P}(y=1)\mathbb{P}(X \in \mathcal{X}_t |\; y = 1) = \alpha L_t$. Replacing $\mathbb{P}(X \in \mathcal{X}_t, y=0)$ by its empirical version n_t/n in (8), we obtain the one-class empirical Gini index

$$i_G^{OC}(t) \quad = \quad \frac{n_t \alpha n L_t}{(n_t + \alpha n L_t)^2}. \tag{9}$$

This one-class index can be seen as a *semi-empirical* version of (8), in the sense that it is obtained by considering empirical quantities for the (observed) inlier behavior and population quantities for the (non-observed) outlier behavior. Now, maximizing the population version of the impurity decrease $\Delta i_G^{theo}(t, t_L, t_R)$ as defined in (7) is equivalent to minimizing

$$p(t_L|t)\ i_G^{theo}(t_L) \ + \ p(t_R|t)\ i_G^{theo}(t_R). \tag{10}$$

Considering semi-empirical versions of $p(t_L|t)$ and $p(t_R|t)$, as for (9), gives $p_n(t_L|t) = (n_{t_L} + \alpha n L_{t_L})/(n_t + \alpha n L_t)$ and $p_n(t_R|t) = (n_{t_R} + \alpha n L_{t_R})/(n_t + \alpha n L_t)$. Then, the semi-empirical version of (10) is

$$p_n(t_L|t)\ i_G^{OC}(t_L) \ + \ p_n(t_R|t)\ i_G^{OC}(t_R) \quad = \quad \frac{1}{(n_t + \alpha n L_t)} \left(\frac{n_{t_L} \alpha n L_{t_L}}{n_{t_L} + \alpha n L_{t_L}} + \frac{n_{t_R} \alpha n L_{t_R}}{n_{t_R} + \alpha n L_{t_R}} \right) \tag{11}$$

where $1/(n_t + \alpha n L_t)$ is constant when the split varies. This means that finding the split minimizing (11) is equivalent to finding the split minimizing

$$I_G^{OC}(t_L, t_R) = \frac{n_{t_L} \alpha n L_{t_L}}{n_{t_L} + \alpha n L_{t_L}} + \frac{n_{t_R} \alpha n L_{t_R}}{n_{t_R} + \alpha n L_{t_R}}. \tag{12}$$

Note that (12) can be obtained from the two-class impurity decrease (4) as described in the naive approach paragraph in Section 3. In other words, it is the naive one-class version of (4).

Remark 3 *(*Direct link with the two-class framework*) The two-class proxy of the Gini impurity decrease (4) is recovered from (12) by replacing $\alpha n L_{t_L}$ (resp. $\alpha n L_{t_R}$) by n'_{t_L} (resp. n'_{t_R}), the number of second class instances in t_L (resp. in t_R). When generating αn of them uniformly on $\mathcal{X}$, $\alpha n L_t$ is the expectation of n'_t .*

As detailed in Section 3.1, this approach suffers from the curse of dimensionality. We can summarize the problem as follows. Note that when setting $n'_t := \alpha n L_t$, the class ratio $\gamma_t = n'_t / n_t$ is then equal to

$$\gamma_t \ = \ \alpha n L_t \ / \ n_t. \tag{13}$$

This class ratio is close to 0 for many nodes t, which makes the Gini criterion unable to discriminate accurately between the (hidden) outliers and the inliers. Minimizing this criterion produces splits corresponding to $\gamma_t \simeq 0$ in Figure 2: one of the two child nodes, say t_L contains almost all the data.

5.2. Adaptive approach

The solution presented Section 3 is to remove the uniform assumption for the outlier class. From the theoretical point of view, the idea is to choose in an adaptive way (*w.r.t.* the volume of $\mathcal{X}_t$) the number αn, which can be interpreted as the number of (hidden) outliers. Doing so, we aim at avoiding $\alpha n L_t \ll n_t$ when L_t is too small. Namely, with γ_t defined in (13), we aim at avoiding $\gamma_t \simeq 0$ when $L_t \simeq 0$. The idea is to consider $\alpha(L_t)$ and $n(L_t)$ such that $\alpha(L_t) \to 1$, $n(L_t) \to \infty$ when $L_t \to 0$. We then define the one-class adaptive proxy of the impurity decrease by

$$I_G^{OC-ad}(t_L, t_R) \;=\; \frac{n_{t_L}\alpha(L_t)\cdot n(L_t)\cdot L_{t_L}}{n_{t_L}+\alpha(L_t)\cdot n(L_t)\cdot L_{t_L}} + \frac{n_{t_R}\alpha(L_t)\cdot n(L_t)\cdot L_{t_R}}{n_{t_R}+\alpha(L_t)\cdot n(L_t)\cdot L_{t_R}}. \qquad (14)$$

In other words, instead of considering one general model One-Class-Model(n, α) defined in Section 5.1, we adapt it to each node t, considering One-Class-Model($n(L_t)$, $\alpha(L_t)$) *before computing the best split.* When growing the tree, using One-Class-Model($n(L_t)$, $\alpha(L_t)$) allows to maintain a non-negligible expected proportion of outliers in the node to be splitted, despite L_t becomes close to zero. Of course, constraints have to be imposed to ensure consistency between these models. Recalling that the number N of inliers is a realization of $\mathbf{N}$ following a Binomial distribution with parameters $(n, 1-\alpha)$, a first natural constraint on $\big(n(L_t), \alpha(L_t)\big)$ is

$$(1-\alpha)n = \big(1-\alpha(L_t)\big)\cdot n(L_t) \qquad \text{for all} \;\; t, \qquad (15)$$

so that the expectation of $\mathbf{N}$ remains unchanged.

Remark 4 *In our adaptive model One-Class-Model($n(L_t)$, $\alpha(L_t)$) which varies when we grow the tree, let us denote by $\mathbf{N}(L_t) \sim Bin\big(n(L_t), 1-\alpha(L_t)\big)$ the* r.v. *ruling the number of inliers. The number of inliers N is still viewed as a realization of it. Note that the distribution of $\mathbf{N}(L_t)$ converges in distribution to $\mathcal{P}\big((1-\alpha)n\big)$ a Poisson distribution with parameter $(1-\alpha)n$ when $L_t \to 0$, while the distribution $Bin\big(n(L_t), \alpha(L_t)\big)$ of the* r.v. *$n(L_t) - \mathbf{N}(L_t)$ ruling the number of (hidden) outliers goes to infinity almost surely. In other words, the asymptotic model (when $L_t \to 0$) consists in assuming that the number of inliers N we observed is a realization of $\mathbf{N}_\infty \sim \mathcal{P}\big((1-\alpha)n\big)$, and that an infinite number of outliers have been hidden.*

A second natural constraint on $\big(\alpha(L_t), n(L_t)\big)$ is related to the class ratio γ_t. As explained in Section 3.1, we do not want γ_t to go to zero when L_t does. Let us say we want γ_t to be constant for all node t, equal to $\gamma > 0$. From the constraint $\gamma_t = \gamma$ and (13), we get

$$\alpha(L_t)\cdot n(L_t)\cdot L_t = \gamma n_t := n_t'. \qquad (16)$$

The constant γ is a parameter ruling the expected proportion of outliers in each node. Typically, $\gamma = 1$ so that there is as much expected uniform (hidden) outliers than inliers at each time we want to find the best split minimizing (14). Equations (15) and (16) allow to explicitly determine $\alpha(L_t)$ and $n(L_t)$: $\alpha(L_t) = n_t' / \big((1-\alpha)nL_t + n_t'\big)$ and $n(L_t) = \big((1-\alpha)nL_t + n_t'\big)/L_t$. Regarding (14), $\alpha(L_t)\cdot n(L_t)\cdot L_{t_L} = \frac{n_t'}{L_t} L_{t_L} = n_t' \frac{\text{Leb}(\mathcal{X}_{t_L})}{\text{Leb}(\mathcal{X}_t)}$ by (16) and $\alpha(L_t)\cdot n(L_t)\cdot L_{t_R} = n_t' \frac{\text{Leb}(\mathcal{X}_{t_R})}{\text{Leb}(\mathcal{X}_t)}$, so that we recover (5).

6. Conclusion

Through a natural adaptation of both (two-class) splitting criteria and majority vote, this paper introduces a methodology to structurally extend RFs to the one-class setting. Our one-class splitting criteria correspond to the asymptotic behavior of an adaptive outliers generating methodology, so that consistency with two-class RFs seems respected. While no statistical guaranties have been derived in this paper, a strong empirical performance attests the relevance of this methodology.

Acknowledgments

This work has been partially supported by the industrial chair ”Machine Learning for Big Data” from Telecom ParisTech.

References

C.C. Aggarwal and P.S. Yu. Outlier detection for high dimensional data. In *ACM Sigmod Record*, 2001.

Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural comp.*, 1997.

V. Barnett and T. Lewis. *Outliers in statistical data*. Wiley New York, 1994.

S. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *KDD*, 2003.

G. Biau and E. Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.

G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(Sep):2015–2033, 2008.

G. Blanchard, C. Schäfer, and Y. Rozenholc. Oracle bounds and exact algorithm for dyadic classification trees. In *International Conference on Computational Learning Theory*, pages 378–392. Springer, 2004.

L. Breiman. Random forests. *Machine Learning*, 2001. ISSN 0885-6125.

M.M. Breunig, H.P. Kriegel, R.T. Ng, and J. Sander. LOF: identifying density-based local outliers. In *ACM Sigmod Rec*, 2000.

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 2009.

S. Clémençon and S. Robbiano. Anomaly Ranking as Supervised Bipartite Ranking. In *ICML*, 2014.

S. Clémençon and N. Vayatis. Tree-based ranking methods. *IEEE Transactions on Information Theory*, 55(9):4316–4336, 2009.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 1995.

Chesner Désir, Simon Bernard, Caroline Petitjean, and Laurent Heutte. One class random forests. *Pattern Recogn.*, 2013.

R. Díaz-Uriarte and S.A. De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 2006.

John HJ Einmahl and David M Mason. Generalized quantile processes. *The Annals of Statistics*, pages 1062–1078, 1992.

E. Eskin. Anomaly detection over noisy data using learned probability distributions. In *ICML*, 2000.

P.A. Flach. The geometry of ROC space: understanding ML metrics through ROC isometrics. In *ICML*, 2003.

Y. Freund, R.E Schapire, et al. Experiments with a new boosting algorithm. In *ICML*, 1996.

R. Genuer, J.-M. Poggi, and C. Tuleau. Random forests: some methodological insights. *arXiv:0811.3619*, 2008.

R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recog. Letters*, 2010.

P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 2006.

C. Gini. Variabilita e mutabilita. *Memorie di metodologia statistica*, 1912.

Sudipto Guha and Okke Schrijvers. Robust random cut forest based anomaly detection on streams. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2712–2721, 2016.

T.K. Ho. The random subspace method for constructing decision forests. *TPAMI*, 1998.

V.J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artif. Intel. Review*, 2004.

KDDCup. The third international knowledge discovery and data mining tools competition dataset. 1999.

M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

F.T. Liu, K.M. Ting, and Z.H. Zhou. Isolation forest. In *ICDM*, 2008.

F.T. Liu, K.M. Ting, and Z-H. Zhou. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data*, March 2012. ISSN 1556-4681. doi: 10.1145/2133360.2133363. URL http://doi.acm.org/10.1145/2133360.2133363.

G. Louppe. Understanding random forests: From theory to practice. *arXiv:1407.7502*, 2014.

M. Markou and S. Singh. Novelty detection: a review part 1: statistical approaches. *Signal proc.*, 2003.

P. Panov and S. Džeroski. *Combining bagging and random subspaces to create better ensembles.* Springer, 2007.

A. Patcha and J.M. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 2007.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *JMLR*, 2011.

W. Polonik. Minimum volume sets and generalized quantile processes. *Stochastic Processes and their Applications*, 1997.

J.A Quinn and M. Sugiyama. A least-squares approach to anomaly detection in static and sequential data. *Pattern Recognition Letters*, 2014.

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*, 2001.

E. Schubert, R. Wojdanowski, A. Zimek, and H-P. Kriegel. On evaluation of outlier rankings and outlier scores. In *SDM*, 2012.

Erwan Scornet, Gérard Biau, Jean-Philippe Vert, et al. Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741, 2015.

C. Scott and G. Blanchard. Novelty detection: Unlabeled data definitely help. In *AISTATS*, pages 464–471, 2009.

C.D Scott and R.D Nowak. Learning minimum volume sets. *JMLR*, 2006.

C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE MC2R*, 2001.

J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis.* Cambridge univ. press, 2004.

T. Shi and S. Horvath. Unsupervised learning with random forest predictors. *J. Comp. Graph. Stat.*, 2012.

M.L. Shyu, S.C. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, DTIC Document, 2003.

M. Sugiyama. Superfast-trainable multi-class probabilistic classifier by least-squares posterior fitting. *IEICE Transactions on Information and Systems*, 2010.

V. Svetnik, A. Liaw, C. Tong, J C. Culberson, R.P Sheridan, and B.P Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *J. Chem. Inf. Model.*, 2003.

M. Tavallaee, E. Bagheri, W. Lu, and A.A. Ghorbani. A detailed analysis of the kdd cup 99 data set. In *IEEE CISDA*, 2009.

D.MJ Tax and R.PW Duin. Uniform object generation for optimizing one-class classifiers. *JMLR*, 2002.

K. Yamanishi, J.I. Takeuchi, G. Williams, and P. Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In *KDD*, 2000.