An Alternative Exploitation of Isolation Forests for Outlier Detection

Antonella Mensi[1](✉), Alessio Franzoni[1], David M. J. Tax[2], and Manuele Bicego[1]

[1] Department of Computer Science, University of Verona, Verona, Italy
{antonella.mensi,manuele.bicego}@univr.it,
alessio.franzoni_01@studenti.univr.it

[2] Faculty of Electrical Engineering, Mathematics and Computer Science, TU Delft, Delft, The Netherlands
D.M.J.Tax@tudelft.nl

Abstract. Isolation Forests are one of the most successful outlier detection techniques: they isolate outliers by performing random splits in each node. It has been recently shown that a trained Random Forest-based model can also be used to define and extract informative distance measures between objects. Although their success has been shown mainly in the clustering field, we propose to extract these pairwise distances between the objects from an Isolation Forest and use them as input to a distance or density-based outlier detector. We show that the extracted distances from Isolation Forests are able to describe outliers meaningfully. We evaluate our technique on ten benchmark datasets for outlier detection: we employ three different distance measures and evaluate the obtained representation using a density-based classifier, the Local Outlier Factor. We also compare the methodology to the standard Isolation Forests scheme.

Keywords: Outlier detection · Isolation forests · Random forest-based similarity

1 Introduction

Isolation Forests (IF) [16,18] represent a Random Forest-based technique for outlier detection, which success have been assessed in many different contexts: for example, in the comparative analysis shown in [9], they were proven to be the most successful methodology to solve this task. In contrast to other Random Forests approaches for outlier detection [7,23], which are based on a standard classification Random Forest trained on normal data and artificially generated outliers, Isolation Forests use trees in which splits are performed completely at random (similarly to the Extremely Randomized Trees [10]). Given the trees, IFs solve outlier detection using the concept of "isolation", which encodes the fact that outliers are probably well separated from the rest, thus being able to be

A. Torsello et al. (Eds.): S+SSPR 2020, LNCS 12644, pp. 34–44, 2021.
https://doi.org/10.1007/978-3-030-73973-7_4

"isolated" from the remainder of the data within the early splits of the tree. Thus, the anomaly degree of a given point can be detected by looking at the depth of the leaf it reaches. Isolation Forests have been extensively employed, extended and improved in many different aspects [8,11,13,14,17,19,24]: most of these extensions [8,11,13,14,17,24] were devoted to improve the training stage, for example by defining novel ways to split a node; few of them focus on improving the testing phase, i.e. the anomaly score [13,19].

In this paper we propose and investigate an alternative exploitation of the Isolation Forests for outlier detection: instead of employing the isolation concept, we investigate the possibility of exploiting the IF to derive pairwise distances between objects, to be then used as input for a distance or density-based outlier detection classifier.

The proposed approach starts from the following observation: Random Forests (RF) are not used solely for classification or regression, but also as a valid and flexible data description tool. For example, in the field of clustering, there are different approaches which exploit the concept that the intrinsic nature of Random Forests allows to describe data in a meaningful way. In all these techniques –the so-called *distance-based RF clustering* methods [3,4,23,26,27]– the idea is to exploit RFs to derive a dissimilarity measure between points, to be subsequently used as input to a distance-based classifier. These measures have been proven to be more descriptive than standard geometric-based distances such as the Euclidean distance, and have been successfully applied in many different domains [1,12,21,22]. In almost all these methods the trained forests are standard binary classification RFs, built using the points to be clustered and a synthetically generated negative class. Very recently [3], however, other learning schemes have been investigated, able to work without generating a synthetic negative class that tends to hide the true nature and complexity of the data. Among other learning strategies, those based on random mechanisms were shown to perform surprisingly well, permitting to derive meaningful and informative distances.

Following these findings, we propose an alternative IF-based outlier detection scheme, in which we exploit Isolation Forests to derive dissimilarities to be used inside a distance-based outlier detector. In the paper we investigated three different strategies for computing the dissimilarity, based on different intuitions [23,27]. To investigate the suitability of the proposed framework we employed ten different benchmark outlier detection datasets, evaluating the different dissimilarities also in comparison with the standard Isolation Forest scheme. Results were encouraging, confirming the richness of the information that can be extracted from this particular type of Random Forests.

The remainder of the paper is divided as follows: in Sect. 2 we present the Isolation Forests in detail; in Sect. 3 we describe the proposed methodology and then we test it in Sect. 4. In Sect. 5 we make some conclusions.

2 Isolation Forests

The most successful and used Random Forest-based technique for outlier detection is called Isolation Forest, or IF [16,18]. Differently from other RF-based methodologies for outlier detection, which create artificial outliers in order to employ RF for classification [7,23], IFs work in a completely unsupervised way. They aim at separating each object from the rest of the dataset, independently of the class it belongs to. The success of the IFs can be attributed to the way in which they are built –the training phase– and secondly, by how the score of each object traversing the forest is computed –the testing one. In the two following Subsections we illustrate in detail such procedures.

2.1 Training Phase

An Isolation Forest is composed of several Isolation Trees (iTrees), which are built using a random subsample of the training set drawn without replacement. Each iTree is built recursively by partitioning each node into two children nodes in a completely random way, inspired by the Extremely Randomized Trees [10]. An axis-parallel split is performed in the following way: a feature is chosen completely at random, and then a random choice is made also for the value along which to split, in the domain of the selected feature. The tree is built until a stopping criterion is met: either we have reached the maximum established depth or it is impossible to split the node.

This tree structure is able to well differentiate outliers from inliers due to the fact that the former are usually fewer, different and heterogeneous with respect to the rest of the dataset. Indeed early splits will have a higher probability to separate outliers from the rest of the data due to the nature of outliers. Therefore we can infer that on average outliers will tend to end up in leaves that have a smaller depth than those that inliers will reach.

2.2 Testing Phase

In the testing phase an object x traverses each tree of a trained IF and a score is inferred, indicating the probability of x being an outlier. The definition of anomaly score $s(x)$, given by Liu et al. [16,18], is as follows:

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

where S is the number of training samples used to build a tree, $c(S)$ is a normalization factor needed for comparing differently built forests and $E(h(x))$ is the average path length across all trees –for a more detailed explanation please refer to [16,18]. The score, which varies in the range between 0 and 1 behaves as expected: a smaller average depth will lead to a higher score which increases the probability of a point to be an outlier.

3 Methodology

The proposed methodology consists of three steps:

1. Train an Isolation Forest model $\mathcal{F}$.
2. Extract from $\mathcal{F}$ a distance matrix $\mathbf{D}$ which contains in cell (x, y) the pairwise distance between the x^{th} and y^{th} object. We call it the IF-distance.
3. Classify the objects using an outlier detector that takes $\mathbf{D}$ as input.

Step 1: Training of IF

The first step represents the standard training of Isolation Forests, as described in Sect. 2. We train a forest $\mathcal{F}$ composed of T trees. Each tree t has been built using S samples drawn without replacement from the training set. The recursive building procedure continues until a maximum depth D is reached. Within each tree t we define the following elements: (i) *root* is the root node of the tree; (ii) n is either an internal node of the tree, i.e. a node which can be split and is not the root, or a leaf node. Each node n contains $< S$ objects: we indicate this quantity with $|n|$ and (iii) $d()$ is the depth function which retrieves the depth of each node, where $d(root) = 0$.

Step 2: Derivation of IF-distance

First, we introduce some useful notation. When objects x and y are traversing a tree t, we define: i) $l_t(x)$ is the leaf node reached by x which has depth $d_t(x) = d(l_t(x))$; ii) $\mathcal{P}_t^x = \{n_1, n_2, \ldots n_{d_t(x)}\}$ is the path traversed by x in t in terms of set of nodes, excluding the root –since it is traversed by all objects. Note that $d_t(x) = |\mathcal{P}_t^x|$. iii) $LCA_t(x, y)$ is the lowest common ancestor of x and y, i.e. the last node in which x and y are together. The split defined in this node will separate x from y; iv) $\lambda_t(x, y) = d(LCA_t(x, y))$ and v) $\mathcal{P}_t^{(x,y)} = \{n_1, ..., LCA_t(x, y)\}$ is the path traversed by both objects, i.e. the subset of nodes traversed by both x and y. Note that $\lambda_t(x, y) = |\mathcal{P}_t^{(x,y)}|$.

The IF-distance $\mathbf{D}$ has been computed using three different proposals, widely and successfully employed in the clustering scenario [23,27].

1. In [4,23] two objects in a tree t are similar if they end up in the same leaf. Therefore, in a forest, two objects are more similar if they reach the same leaf in a greater number of trees. Formally, given objects x and y the *Shi* similarity between the two objects is defined as:

$$SimShi(x, y) = \frac{\sum_{t \in \mathcal{F}} \mathbb{1}(l_t(x) = l_t(y))}{T} \tag{2}$$

where $\mathbb{1}$ is the indicator function that returns 1 if the two leaves are equal and T is the number of trees in $\mathcal{F}$. This measure is then transformed into a distance in the following way:

$$Shi(x, y) = \sqrt{1 - SimShi(x, y)}. \tag{3}$$

The other two measures are defined by [27]. The authors generalize the concept introduced by [22]: objects which do not arrive at the same leaf may share some similarity, that can be measured via the length of their common path. The novel measures introduced in [27] are *ClustRF-Strct-Unfm* and *ClustRF-Strct-Adpt* which we will call *SimZhu2* and *SimZhu3* for the sake of simplicity:

2. Given two objects x and y that traverse a tree t, $SimZhu2_t$ is defined as:

$$SimZhu2_t(x,y) = \frac{\lambda_t(x,y)}{\max\{|\mathcal{P}_t^x|, |\mathcal{P}_t^y|\}}. \tag{4}$$

   The length of the common path is divided by the length of the longest path: this is necessary since, given a fixed λ, the similarity between x and y should be higher if the denominator is closer to λ. The measure is extended to $\mathcal{F}$ in the following way:

$$SimZhu2(x,y) = \frac{\sum_{t\in\mathcal{F}} SimZhu2_t(x,y)}{T} \tag{5}$$

   which is simply the average similarity between the two objects. We transform the similarity into a distance as follows:

$$Zhu2(x,y) = 1 - SimZhu2(x,y). \tag{6}$$

3. The variant called *SimZhu3* is a weighted version of *SimZhu2*. Each node is considered to have a depth-based importance since objects which are together in a very deep node are more similar than objects which are together only, for example, in the root. To account for this, in [27] they define the weight of a node k to be $\frac{1}{|k|}$ since smaller nodes are usually deeper in a tree. Therefore given objects x and y the similarity $SimZhu3_t$ in a tree t is:

$$SimZhu3_t(x,y) = \frac{\sum_{k\in\mathcal{P}_t^{(x,y)}} \frac{1}{|k|}}{\sum_{k\in\mathcal{P}_t^b} \frac{1}{|k|} + \frac{1}{|l_t(b)|}} \tag{7}$$

   where $b = \underset{x,y}{\operatorname{argmax}} |\mathcal{P}_t^b|$. The measure is extended to $\mathcal{F}$ in the following way:

$$SimZhu3(x,y) = \frac{\sum_{t\in\mathcal{F}} SimZhu3_t(x,y)}{T}. \tag{8}$$

   We transform the similarity into a distance as follows:

$$Zhu3(x,y) = 1 - SimZhu3(x,y). \tag{9}$$

Step 3: Distance-based outlier detection

After having computed $\mathbf{D}$, we can apply any distance-based outlier detection method. Different techniques exist in the literature –for a detailed explanation please refer to [6]. The most simple methods exploit the distance to the k^{th}

neighbor in different ways: an example is *NNd* [25]. NNd states that if the distance between an object and its nearest neighbor is greater than the distance between the latter and its nearest neighbor, then the object under analysis has an increased probability of being an outlier.

Then there are more refined techniques which employ an estimation of the relative density to solve the task, such as the *Local Outlier Factor* (LOF) [5]. LOF works by comparing the neighborhood density of the object under analysis with that of its neighbors. The object has a higher probability of being an outlier if at least one of the neighbors has a denser neighborhood than its own. The classifier has only one parameter to set: K, the neighborhood size. In our work we employ LOF since it is more sophisticated than NNd.

4 Experimental Evaluation

In this Section we first describe the datasets and some experimental details and then we present the obtained results and compare the methodology to the IFs.

4.1 Experimental Details

We evaluate the methodology on 10 UCI ML datasets[1] which were transformed into outlier detection datasets: in all of them nominal attributes were removed. Then the outlier and inlier classes are defined based on previous works: for Breastw, Ionosphere, Pima and Satellite see [16], for Glass and WBC see [15], for Arrhythmia and Wilt refer to [11], for Musk follow [2] and for Letter refer to [20] –as to this dataset further modifications were made other than defining the classes[2]. In Table 1, datasets are described in terms of number of objects, number of features and percentage of outliers. These datasets cover a large range of situations: they differ greatly in the number of features (from 5 up to 164), in the outlier percentage (from 3.17% up to 45.80%) and in the size (the smallest one has 213 samples while the biggest 6435).

After a preliminary evaluation –not shown here–, we chose the following parameters for the IF training: $S = 256, D = \log_2(S), T = 150$. The parameters of the methodology are very easy to set, as shown in [16,18]: indeed we only varied the forest size with respect to the default parametrizations since it shows better performances. Each experiment was repeated 20 times. For each iteration 50% of the objects was randomly assigned to the training set and the other 50% to the testing set, where the former did not contain any outlier.

Given the trained forests, we computed the IF-distances with the three variants described in Sect. 3: *Shi*, *Zhu2* and *Zhu3*. As to the chosen classifier, LOF, after preliminary analyses not shown here, we set $K = 14$ since it allows to achieve the best performances on average. As accuracy measure, as often done in outlier detection, we use the Area under the ROC Curve (AUC).

[1] Available at https://archive.ics.uci.edu/ml/index.php.

[2] All datasets adequately processed can be found at http://odds.cs.stonybrook.edu/, except for Arrhythmia for which we use a different version [11].

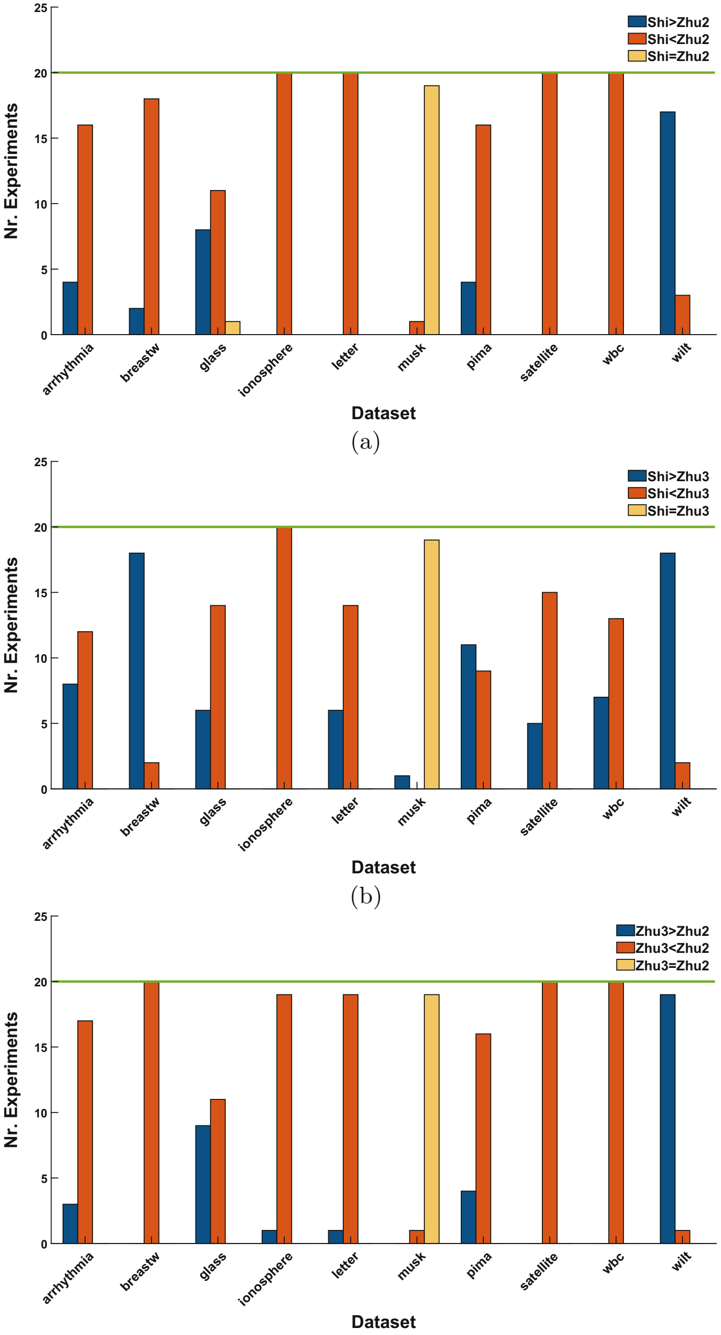
Table 1. Overview of the 10 datasets used for the experimental evaluation.

Datasets	Nr. of objects	Nr. of features	Outlier %
Arrhythmia	452	164	45.80%
Breastw	683	9	34.99%
Glass	214	9	4.21%
Ionosphere	351	32	35.90%
Letter	1600	32	6.25 %
Musk	3062	166	3.17%
Pima	768	8	34.90%
Satellite	6435	36	31.64%
WBC	378	30	5.56 %
Wilt	4839	5	5.39%

4.2 Results

The first analysis compares the three IF-distance measures we can compute from the Isolation Forests. In Figs. 1 (a), (b) and (c) we present a pairwise comparison between the distances: for each dataset we count for how many experiments the first named measure is better than the second (blue bar), the second is better than the first (orange bar) and for how many experiments the two distance measures perform the same (yellow bar). The green line represents the maximum number of experiments per dataset, which is 20. In Figs. 1 (a) and (c) we compare *Zhu2* with the other distances: its superiority is straightforward. Indeed for all datasets except two it is better than the other distance measures and for one, Musk, the performances are equal. Comparing instead *Zhu3* and *Shi*, in Fig. 1 (b), we can observe that in many cases *Zhu3* is the better choice, except for Breastw, Pima and Wilt which are all rather small; for Musk we observe again that the performances are independent of the used distance measure.

The second analysis compares the proposed methodology with the IF: we present the results in Table 2. For each dataset we report the median across the 20 repetitions. In detail, as to the proposed methodology we report the accuracy achieved with the best distance measure, which is indicated between parenthesis–if *All* is present, it means that all distances lead to the same accuracy. In addition we performed a Wilcoxon signed-rank test to assess whether the differences between the methodologies are statistically significant. The scores in bold are the best ones, and if a * is present, then the difference with the other methodology is statistically significant. From Table 2 we can observe that on seven datasets the best accuracy is reached when using the proposed methodology. In detail in four cases it is achieved when using *Zhu2* as distance measure and for four out of these seven datasets the difference is statistically significant –for Wilt and Letter the improvement is remarkable. In addition even though for the remaining datasets IF is significantly better, only for Breastw the proposed methodology actually fails. Finally if we observe the average results across all



(a)

(b)

(c)

Fig. 1. Comparison between the proposed distances. Respectively each figure compares (a) Shi with Zhu2 (b) Shi with Zhu3 and (c) Zhu3 with Zhu2.

Table 2. Accuracy comparison between the IF and the proposed methodology.

Dataset	IF	LOF (Best Dist.)
Arrhythmia	0.773	**0.778(Shi)**
Breastw	**0.995***	0.582(Zhu2)
Glass	0.729	**0.733(Zhu2)**
Ionosphere	0.894	**0.906(Zhu2)**
Letter	0.641	**0.861*(Zhu2)**
Musk	0.988	**1.000*(All)**
Pima	**0.738***	0.696(Zhu2)
Satellite	0.810	**0.840*(Zhu2)**
Wbc	**0.954***	0.938(Zhu2)
Wilt	0.516	**0.903*(Shi)**
Average	0.804	**0.824**

the datasets the maximum accuracy is reached with the proposed technique. We can thus conclude it is advantageous to employ the IF-distance: this is particularly true if the dataset is big enough, i.e. if it has > 1000 objects.

5 Conclusions

In this paper we propose a novel methodology for outlier detection that exploits Isolation Forests. From the latter we extract a distance matrix which is then input to an outlier detector: the novel representation should be able to meaningfully describe the objects and identify the outliers, thanks to the intrinsic nature of the trees composing the forest. We employed different RF-based distance measures and evaluate the methodology on ten datasets: the proposed technique has been proven to be advantageous with respect to using Isolation Forests alone.

References

1. Abba, M.C., et al.: Breast cancer molecular signatures as determined by sage: correlation with lymph node status. Mol. Cancer Res. **5**(9), 881–890 (2007)
2. Aggarwal, C.C., Sathe, S.: Theoretical foundations and algorithms for outlier ensembles. SIGKDD Explor. Newsl. **17**(1), 24–47 (2015)
3. Bicego, M., Escolano, F.: On learning random forests for random forest-clustering. In: Proceedings of the 25th International Conference on Pattern Recognition, Forthcoming (2021)
4. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
5. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: Proceedings of SIGMOD International Conference on Managing Data, pp. 93–104 (2000)

6. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**(3), 1–58 (2009)
7. Désir, C., Bernard, S., Petitjean, C., Heutte, L.: One class random forests. Pattern Recogn. **46**, 3490–3506 (2013)
8. Ding, Z., Fei, M.: An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. IFAC Proc. **46**(20), 12–17 (2013)
9. Emmott, A.F., Das, S., Dietterich, T., Fern, A., Wong, W.K.: Systematic construction of anomaly detection benchmarks from real data. In: Proceedings of SIGKDD Workshop Outlier Detection and Description, pp. 16–21 (2013)
10. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006)
11. Goix, N., Drougard, N., Brault, R., Chiapino, M.: One class splitting criteria for random forests. In: Proceedings of 9th Asian Conference Machine Learning, vol. 77, pp. 343–358 (2017)
12. Gray, K.R., Aljabar, P., Heckemann, R.A., Hammers, A., Rueckert, D.: Random forest-based similarity measures for multi-modal classification of Alzheimer's disease. NeuroImage **65**, 167–175 (2013)
13. Guha, S., Mishra, N., Roy, G., Schrijvers, O.: Robust random cut forest based anomaly detection on streams. In: Proceedings of the 33rd International Conference on Machine Learning, vol. 48, pp. 2712–2721 (2016)
14. Hariri, S., Kind, M.C., Brunner, R.J.: Extended isolation forest (2018). arXiv:1811.02141
15. Keller, F., Muller, E., Bohm, K.: HICS: high contrast subspaces for density-based outlier ranking. In: IEEE International Conference on Data Engineering, pp. 1037–1048. IEEE (2012)
16. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: IEEE International Conference on Data Mining, pp. 413–422 (2008)
17. Liu, F.T., Ting, K.M., Zhou, Z.H.: On detecting clustered anomalies using sciforest. In: ECML PKDD, pp. 274–290 (2010)
18. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data **6**(1), 1–39 (2012)
19. Mensi, A., Bicego, M.: A novel anomaly score for isolation forests. In: International Conference on Image Analysis and Processing, pp. 152–163 (2019)
20. Micenková, B., McWilliams, B., Assent, I.: Learning outlier ensembles: the best of both worlds-supervised and unsupervised. In: Proceedings of SIGKDD Workshop on Outlier Detection and Description, pp. 51–54 (2014)
21. Rennard, S., et al.: Identification of five chronic obstructive pulmonary disease subgroups with different prognoses in the eclipse cohort using cluster analysis. Ann. Am. Thorac. Soc. **12**(3), 303–312 (2015)
22. Shi, T., Seligson, D., Belldegrun, A., Palotie, A., Horvath, S.: Tumor classification by tissue microarray profiling: random forest clustering applied to renal cell carcinoma. Modern Pathol. **18**, 547–557 (2005)
23. Shi, T., Horvath, S.: Unsupervised learning with random forest predictors. J. Comput. Graph. Stat. **15**, 1–21 (2006)
24. 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)
25. Tax, D.: One-class classification; concept-learning in the absence of counter-examples. Ph.D. thesis, Delft University of Technology (2001)

26. Ting, K., Zhu, Y., Carman, M., Zhu, Y., Zhou, Z.H.: Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure. In: Proceedings of International Conference on Knowledge Discovery and Data Mining, pp. 1205–1214 (2016)
27. Zhu, X., Loy, C., Gong, S.: Constructing robust affinity graphs for spectral clustering. In: Proceedings of International Conference on Computer Vision and Pattern Recognition, pp. 1450–1457 (2014)