**CS 622, Introduction to Machine Learning Project Report**

Outlier/Anomaly Detection

**Members:**

* **Sayyed Ahmad Naghavi Nozad (NSHE ID: 2001868645)**
* **Abdul Tauqeer (NSHE ID: 2002154060)**

Table of Contents

[1 Problem definition 3](#_Toc102398573)

[2 Competing methods 4](#_Toc102398574)

[2.1 Local Outlier Factor (LOF): 4](#_Toc102398575)

[2.2 K Nearest Neighbors (kNN): 5](#_Toc102398576)

[2.3 Isolation Forest (*i*Forest) 5](#_Toc102398577)

[2.4 Histogram-Based Outlier Score (HBOS) 7](#_Toc102398578)

[3 Experiment design 7](#_Toc102398579)

[3.1 Evaluation metrics 7](#_Toc102398580)

[3.2 Data description 8](#_Toc102398581)

[3.3 Parameterization 8](#_Toc102398582)

[3.3.1 LOF 9](#_Toc102398583)

[3.3.2 kNN 11](#_Toc102398584)

[3.3.3 *i*Forest 13](#_Toc102398585)

[3.3.4 HBOS 15](#_Toc102398586)

[3.4 Overall analysis 17](#_Toc102398587)

[Figure 1. A simple example of outliers in a simple 2D data [3] 3](#_Toc102398588)

[Figure 2. Illustration of local reachability distance with k=3 in LOF [9] 4](#_Toc102398589)

[Figure 3. LOF scores as visualized by circles around every point. The larger the circle, the bigger the outlier score [8]. 5](#_Toc102398590)

[Figure 4. Isolating operation in iForest done for an inlier in (a) and for an outlier in (b) [12] 6](#_Toc102398591)

[Figure 5. An example of isolation tree in which outliers are potentially easier to spot on [15] 6](#_Toc102398592)

[Figure 6. Accuracy results for LOF over 9 datasets w.r.t. 5 different values for k 10](#_Toc102398593)

[Figure 7. LOF execution time results over 9 datasets and various k values 11](#_Toc102398594)

[Figure 8. AP results over different datasets with three different outlier scoring strategies 12](#_Toc102398595)

[Figure 9. kNN execution time over various values of k 13](#_Toc102398596)

[Figure 10. Accuracy results of iForest over the variation of the number of base estimators (t) 14](#_Toc102398597)

[Figure 11. iForest execution time results for different t values 15](#_Toc102398598)

[Figure 12. HBOS accuracy variation over the number of bins 16](#_Toc102398599)

[Figure 13. HBOS execution time outcomes over the alteration of the bins count 17](#_Toc102398600)

[Figure 14. Three different average accuracy measures over 9 input datasets 18](#_Toc102398601)

[Figure 15. Average Precision (AP) results for all methods over the 9 datasets 18](#_Toc102398602)

# Problem definition

In data analysis, outliers or anomalies (or even in some cases, novelties) are those rare items that deviate significantly from a well-known notion of normal behavior. Such examples may cause suspicions that these data samples might be deliberately created out of a discrepant mechanism or not be in harmony with the remainder of data. Outlier detection refers to methodologies that aim at discovering such rare and exceptional examples in data [1]. Figure 1 manifests a simple example of outliers with two normal clusters in a 2D space. o1, o2, and O3 are all outliers though O3 is a collective outlier [2] in which a small set of points are deviating substantially from the norm.

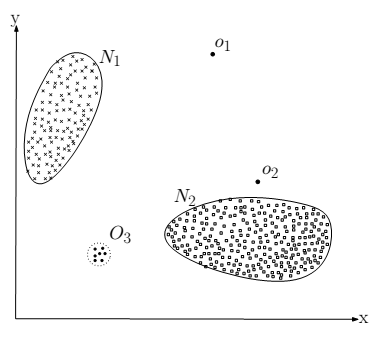


Figure . A simple example of outliers in a simple 2D data [3]

Discovering outliers is sometimes compared with trying to locate a needle in a haystack due the very low rate of outliers existence in every dataset. Outliers can hurt the learning procedure in many applications thus eliminating them is considered an indispensable task. However, in certain cases, outliers could become interesting to the data analyst; such applications are but not limited to controlling cellular phone activity to detect fraudulent usage like stolen phone airtime, detecting fraudulent transactions on credit cards, and so on. Besides, outlier/anomaly detection methods could be considered as a preprocessing step for other advanced data mining tasks to free the input data from unnecessary and erroneous data elements for better and more accurate and reliable performance [3].

Outliers can be separated into two general divisions; local outliers and global outliers. Global outliers are those isolated elements which can be even pointed out with naked eyes if the dataset was capable of being depicted in two dimensions in any way. Local outliers, on the other hand, are deviating from the norm of a distinct cluster/class and hence finding them is much more cumbersome than the global counterparts. Outlier/anomaly detection methods can be, from some specific aspects, divided into the following categories: probabilistic methods, proximity-based methods, graph-based methods, information-theoretic methods, and isolation-based methods. In the following section, we will discuss on some of well-known anomaly identification techniques which are more compared in the experimental evaluations [4], [5].

# Competing methods

There are numerous invented and analyzed outlier detection approaches in the literature, though, here, we are focusing on some of them which are widely used as benchmark methods and all are unsupervised methods. To be noted, regarding our investigations, most of the well-known anomaly detection methods are unsupervised. The famous and useful PyOD library [6] which is used in this project too, and also the PyGOD library [7] approve that. The employed techniques here are as comes next.

## Local Outlier Factor (LOF):

Local Outlier Factor or in short, LOF [8], computes the local density fluctuations around every data point concerning its vicinity. For this matter, first of all, the k-nearest-neighbors of every point are located. This kNN set will proceed into defining the final outlier scores for every point. It should be mentioned that the number of points in this kNN could be more than k, as there could be more than one point standing at the same k-th distance.

Thereafter, for every point w.r.t. its kNN distances, a local reachability distance is delineated which corresponds to whether the point resides in the kNN neighborhood of its surrounding points. The local reachability density is acquired out of such local reachability distances. The farther the point is from its vicinity, the higher the local reachability distances of it from its kNN set, and hence the lower its local reachability density will be. Figure 2 is an example of local reachability distance with k=3, in which for outlier A, the k-distance neighborhood and the subsequent local reachability distances are higher than other inliers.

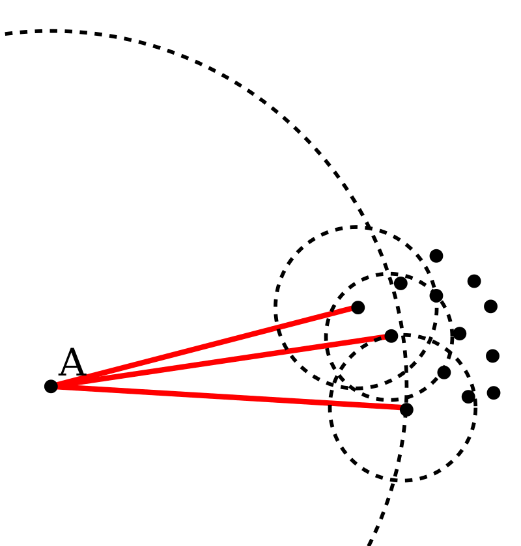


Figure 2. Illustration of local reachability distance with k=3 in LOF [9]

Finally, Local outlier factor for a data element will be the average of local reachability densities of its kNN set over the one belonging to itself. In other words, when the local density of a point is much smaller than its ambient points, the more chance for it to be announced as a latent anomaly. Figure 3 demonstrates a dataset with two clusters and a bunch of outliers around them. For every point, a red circle is plotted around it with a radius correspondent to its outlier degree. Some outliers are both local and global, although this does not make any difference for LOF to calculate the scores.

|  |
| --- |
| Figure . LOF scores as visualized by circles around every point. The larger the circle, the bigger the outlier score [8].  As it is evident, the only prominent parameter here is k, the number of neighbors to be considered for every point in data. However, unfortunately, this parameter does not follow any clear principle as it is stipulated in the original paper, and defining it is another line of research by itself. In our experiments, we follow the suggested value by many sophisticated authors, equal to 20, which also shows practically good performance on real-world datasets. |

## K Nearest Neighbors (kNN):

As in LOF that at the first place, the k nearest neighbors of every data element are discovered, in kNN, the same procedure is carried out at first, though with some more efficient algorithms like in [10], [11]. After finding the kNN set for all points, without a complex investigation, only the distances in such sets are utilized in different strategies to form outlier scores. The more common three strategies to define such scores are using the largest distance in kNN set for every point, the average distance, or the median distance. In some works, the sum of all distances in kNN set is introduced as the anomalous degree for every instance in the input data.

## Isolation Forest (*i*Forest)

Isolation-based idea for outlier detection was initially introduced in [12], [13] and it also follows the ensemble learning method in random forests [14] mostly exerted in classification tasks. In this technique, first, multiple random samplings are carried out with a predefined sample size. Then, upon such random samples, the isolation trees are built by which for every point, the isolation degree is acquired. This isolation degree has a reversed relationship with the final anomaly score. The easier the point is capable to become isolated, the higher the outlierness degree will be designated to that point.

For the purpose of building the isolation trees over the random samples of data, at every level of tree for every internal node, a random attribute with a random split point in the interval of minimum and maximum values of that attribute are chosen. Then two daughter nodes are branched out of such node and the procedure goes on until in a resulting node, either there is only one instance or all points have the same value such that the breaking operation cannot keep running anymore. Such a node uncapable of getting more split is called an external node. Figure 4 illustrates a situation of isolation for an inlier and outlier. In Figure 5, a sketch of isolation tree is depicted in which the path length for isolating an outlier is substantially shorter.

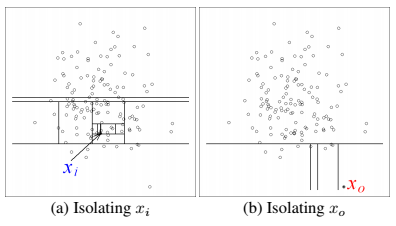


Figure . Isolating operation in iForest done for an inlier in (a) and for an outlier in (b) [12]

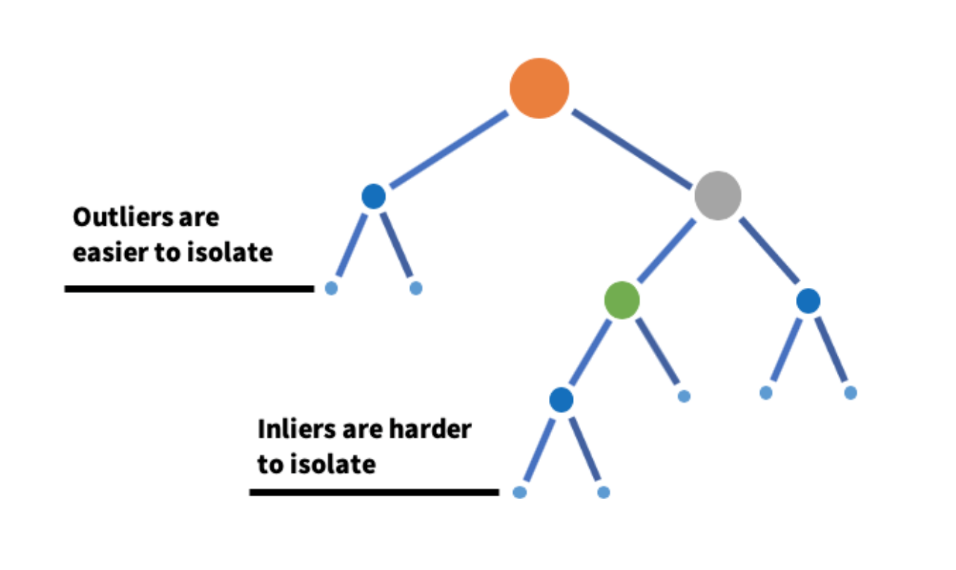


Figure 5. An example of isolation tree in which outliers are potentially easier to spot on [15]

In such isolation trees — which are mostly similar to binary search trees (BST) — real outliers have a higher probability than inliers to become stuck in an external node with a shorter traversing distance from the root. Thus, the isolation proceeding for outliers is considerably briefer than inliers. After gaining the average path lengths for isolating various points in data regarding different isolation trees, by using the harmonic number (Euler’s constant), the final outlier scores are computed as by which one can decide on the degree of anomalousness of every point.

Equation shows the way the final outlier scores are computed. is the path length for data element in a specific isolation tree, while is the average path length for this data element over all isolation trees. in Equation is the average path length of unsuccessful search in BST where is the harmonic number and it can be estimated through the Euler’s constant like in Equation .

## Histogram-Based Outlier Score (HBOS)

Histogram-Based Outlier Detection algorithm, or in brief, HBOS [16], is an unsupervised method capable of working on mixed-type data with both numerical and categorical features, and with a linear time complexity for finding mostly global outliers in a dataset. Its performance on detecting local outliers is poor, though for the purpose of global outliers, it operates multiple times faster than clustering- and nearest-neighbor-based outlier algorithms. The reason behind such speed is its strong assumption on the independence amid data attributes unlike the multivariate approaches, although this happens at the cost of a lower accuracy in general.

The process in this method goes like at the beginning, for every single feature, a univariate histogram is fabricated through either the static bin-width strategy or preferably the dynamic bin-width strategy. The reason that in outlier detection algorithms the dynamic approach for creating the bins is preferred, is the large gaps in value ranges of real-life recorded data. As the outlier detection methods are sensitive to such gaps, thus it would be better to avoid using static-bin approach in which there might be several significant gaps among value boundary of an attribute not at the extreme sites; this could cause unknown or long-tailed distributions to be histogramed not in a sensible way.

Finally, after building the histograms for each feature and a normalization process on the bins, according to Equation , for every data element p, an outlier score is defined concerning the heights of the related bins the element resides in; denotes the related bin’s height and d is the number of variables in data. It is worth mentioning that the only important parameter for this method is the number of bins which is according to a usual rule of thumb set to the square root of the number of total instances in the input data. However, a count equal to 10 is also suggested which works well in various situations.

# Experiment design

In this section, the datasets employed in this project are explained in detail, and also the way they have been preprocessed for outlier detection problem is clarified. Furthermore, for every method, a set of experiments is run for finding the optimal parameter values, although at the end, we will observe that the suggested default values for any of them are working fine on various datasets in this experimentation. Furthermore, the time consumption for every method over different parameter quantities is reported. Finally, with regard to the optimal parameters, the final evaluation outcome in decent bar and radar graph representations will be depicted which is interesting to the eye of the data analyst.

## Evaluation metrics

For the functionality assessment of different methods in this small survey, regarding the stochastic behavior of all mentioned methods, we prefer to use the following metrics widely utilized in the literature: AUROC or area under receiver operating characteristic/curve, AUPRC or area under precision-recall curve, and P@n or precision at rank n [5], [17]. For computing AUROC and AUPRC — which is also called average precision (AP) regarding the specific way of calculating the are under the curve —, we need to work with true positive rate (TPR) or Recall, false positive rate (FPR), and Precision w.r.t. the following formulas:

To calculate P@n, an approximate number for top outliers in data is required from the user, and then among those top candidates, it is examined what number of points are truly designated as anomaly. Such ratio, like AP cannot gain high values due to the imbalance of datasets containing anomalies.

## Data description

In this experimentation, we are utilizing 9 different small and medium sized datasets adopted from the ODDS repository [18], a very much appreciated effort finished in the Stony Brook University, New York. Most of data in this repository are first approved by the famous UCI Machine Learning Repository [19], or the OpenML repository [20], or some other alike repositories. Then, regarding the nature of every dataset, they have preferred to choose some classes and combine them to form the inliers and the same goes for some other minority classes to be designated as outliers. Further descriptions could be found on their website. Some of the properties of these datasets, including the cardinality (#n), dimensionality (#d), number of outliers (#o), and the outliers percentage (%o) are mentioned in Table 1.

Table . Properties of the datasets employed in our experimentation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | #n | #p | #o | %o |
| Arrhythmia | 452 | 274 | 66 | 14.6 |
| Glass | 214 | 9 | 9 | 4.21 |
| Ionosphere | 351 | 33 | 126 | 35.9 |
| Lympho | 148 | 18 | 6 | 4.05 |
| Pima | 768 | 8 | 268 | 34.9 |
| Vertebral | 240 | 6 | 30 | 12.5 |
| Vowels | 1456 | 12 | 50 | 3.43 |
| WBC | 378 | 30 | 21 | 5.56 |
| Wine | 129 | 13 | 10 | 7.75 |

It is evident that only for the Pima and Ionosphere datasets, the percentage of outliers is about one third of the data size which is reflected as a significant amount, while in others it is generally lower than 10%. The Arrhythmia dataset possesses the largest number of dimensions amid all and it could potentially be considered a high-dimensional data which suffers from a non-proportionate data size.

## Parameterization

For setting the appropriate parameters for the four competing methods, as noted above, we prefer to use the default values suggested by the authors. However, in the following, a report on testing different values for the most important parameters is provided and it proves that the suggested parameter values work fine enough on various real situation in this experimentation.

### LOF

For LOF, we run experiments regarding different values for the number of nearest neighbors employed in the method to find the kNN set of every point. We set k equal to 5 values from 20 to 60 with a step length of 10. In subsequent, the execution time outcomes are computed and all these are depicted in Figure 6 and Figure 7. It is evident that LOF does not show so much variation over its performance by increasing the value of k. Just in the case of vertebral in AUROC, lympho in AP, and lympho and glass in P@n, there are some slight variations that could easily be neglected. Moreover, the execution time results in Figure 7 approves that with different quantities for k, there is no significant change in the final execution time for LOF. This is due to the fact that for finding the kNN set for every point, we should calculate its distance to the rest of data which does not care about k at all.

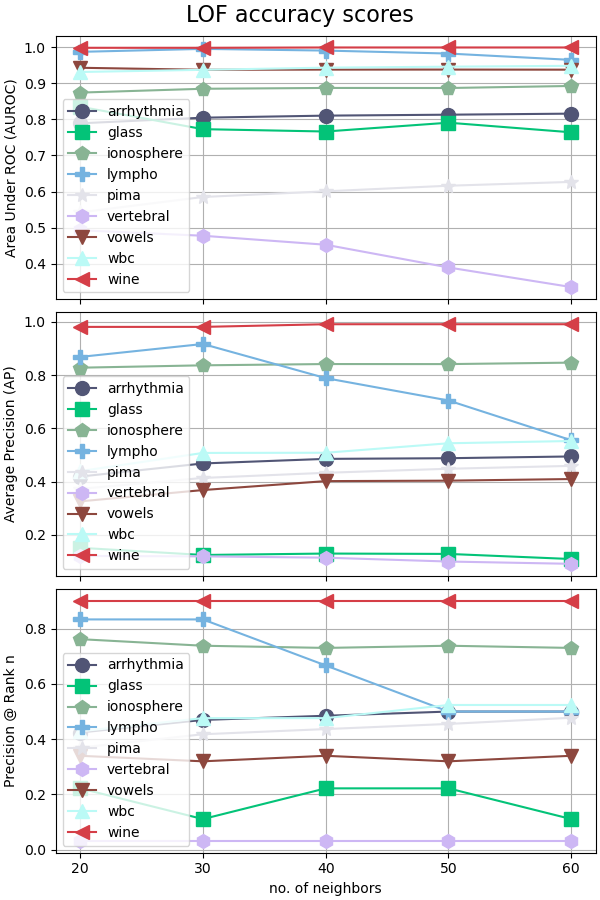


Figure . Accuracy results for LOF over 9 datasets w.r.t. 5 different values for k

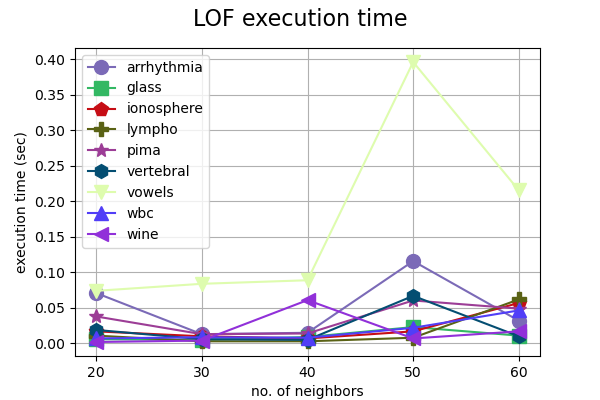


Figure . LOF execution time results over 9 datasets and various k values

### kNN

For kNN, it goes the same as for LOF. The important parameter is the number of neighbors for which we choose the closed interval [5, 25] with the step length of 5. Three different outlier scoring strategies are analyzed and in all of them, average precision (AP) is reported. Figure 8 demonstrates the results in which no significant variation could be observed neither for any of the utilized datasets nor any of three different kNN detectors, including the largest distance, distances mean, and distances median. The execution time graph in Figure 9 works the same way as for LOF in which nothing is changing sensitively w.r.t. parameter k, the number of neighbors in the vicinity of each point.

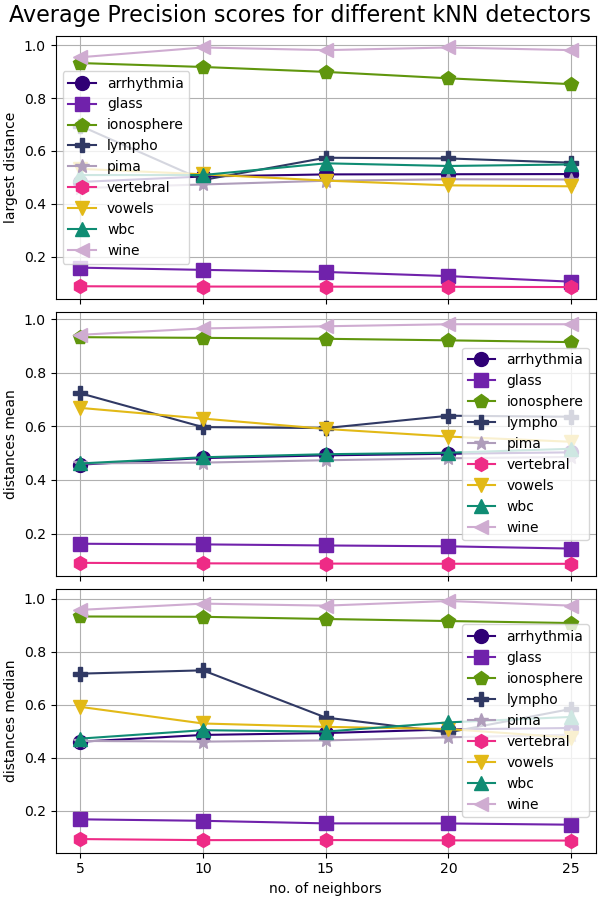


Figure . AP results over different datasets with three different outlier scoring strategies

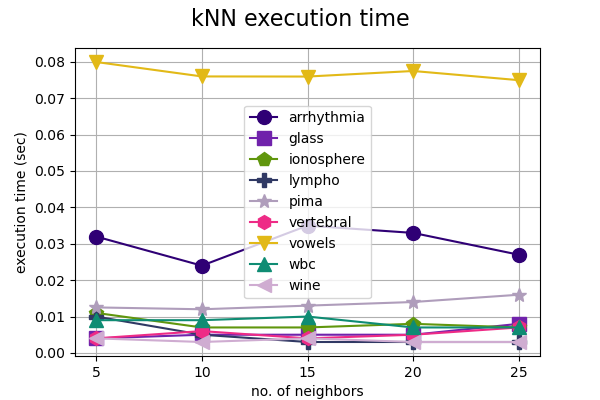


Figure . kNN execution time over various values of k

### *i*Forest

For isolation forest or *i*Forest method, there are two major parameters for which some optimal values have been proposed by the authors. These optimal parameter values work really satisfactory on almost every input dataset with different dimensionalities. One is the number of subsamples (*t*) that should be drawn out of the input data to build the isolation trees, and the other is the size of each subsample (*ψ*). Regarding the applied experiments in the original papers, the number of subsamples or the same base estimators is suggested to be 100, and the second parameter is usually set to 256 or the dataset size if it has a lower cardinality than this suggested quantity. In our analysis, we preferred to vary the first parameter, the number of base estimators, from 100 to 1000 with a step length of 100, to see what the behavior variation of *i*Forest will be.

The accuracy results are depicted in Figure 10, and Figure 11 demonstrates the consumed time by *i*Forest over the fluctuation of parameter *t*. It is evident that the accuracy results are all somewhat constant when the number of base estimators is changing, while the execution time is increasing monotonically w.r.t. to *t* for all datasets. This approves the claim of the authors over the input parameters.

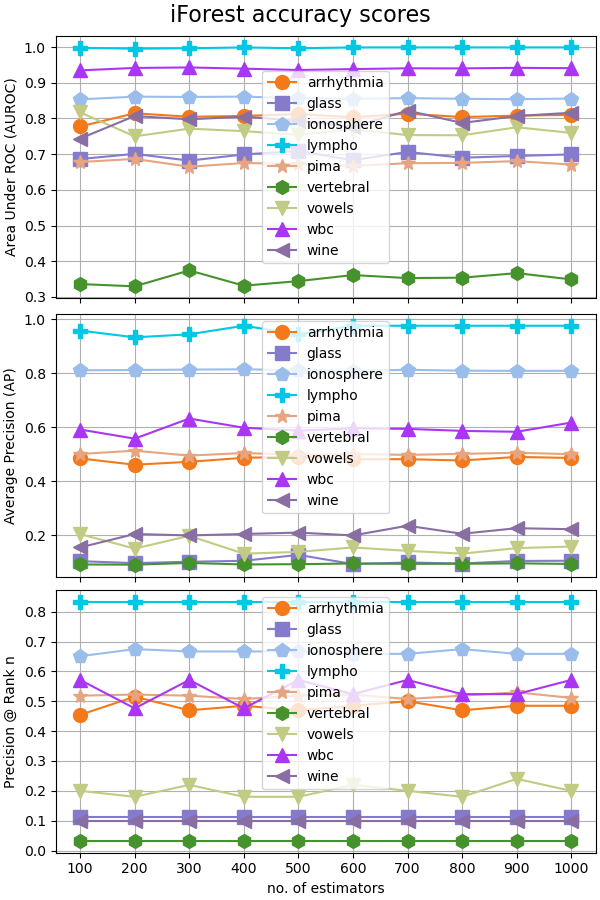


Figure . Accuracy results of iForest over the variation of the number of base estimators (t)

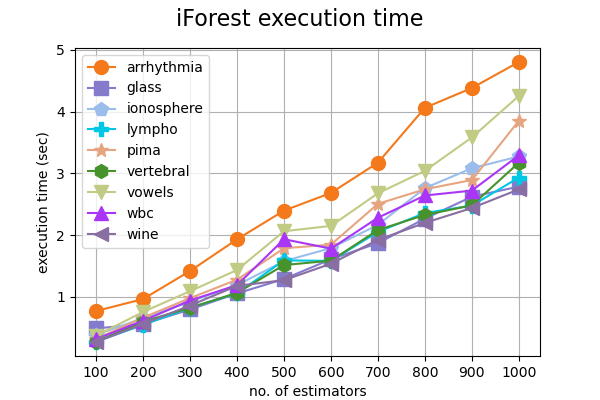


Figure . iForest execution time results for different t values

### HBOS

For this simple yet very fast method, there is only one important parameter which should be taken into account. It is the number of bins for building the histograms over every individual attribute in data, and here, we change it from 5 to 25 with a step length of 5. The accuracy and execution time results are presented respectively in Figure 12 and Figure 13, and it is clear that none of them shows a specific behavior over the variation of the bins count.

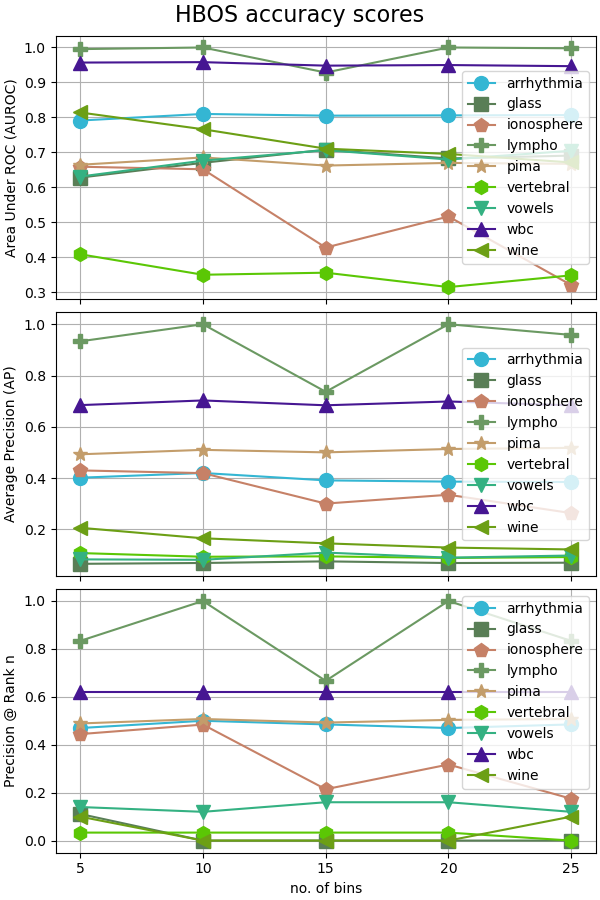


Figure . HBOS accuracy variation over the number of bins

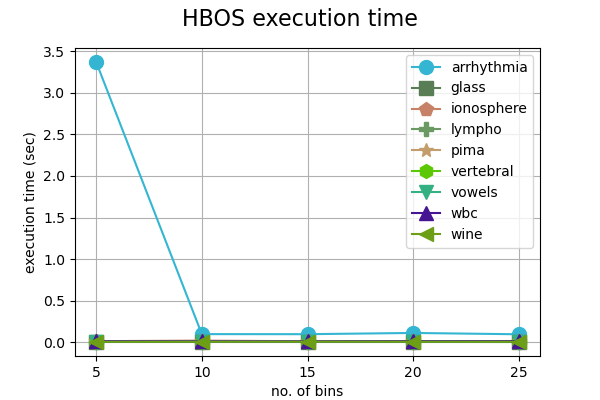


Figure . HBOS execution time outcomes over the alteration of the bins count

## Overall analysis

Here, we have chosen the accuracy results belonging to the suggested parameter values by the authors in every original paper, and illustrate them in two different graphs as follows. The suggested parameters are as comes next: k or the number of nearest neighbors, equal to 20 and 5 for LOF and kNN respectively; *t* and *ψ* equal to, in order, 100 and 256, for *i*Forest; and number of bins equal to 10 for HBOS.

Figure 14 demonstrates the bar graph for the average accuracy values over 9 input datasets, and it is obvious that for all the utilized data in our experimentation, average performance is almost the same and not poor overall. Figure 15, on the other hand, shows the Average Precision (AP) outcomes over the 9 input datasets individually in a decent radar graph; the best advantage of such graphs may be that the larger the area belonging to the container of a specific method, the better the performance it will have. It is apparent from this graph that kNN stands at the top though not with a significant distance from the other methods.

It is also worth mentioning that for AP, exceptionally, even in the best case, the final result for outlier detection methods may not be a remarkable value like over 0.8. This happens naturally due to the class imbalance in outlier datasets where the ratio of outliers is much lower than inliers, and more importantly, the Precision measure exerted in AP which unlike the AUROC takes the false positive rate (FPR) into account too [21].

Figure . Three different average accuracy measures over 9 input datasets

Figure . Average Precision (AP) results for all methods over the 9 datasets

[1] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artif Intell Rev*, vol. 22, no. 2, pp. 85–126, 2004.

[2] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[3] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[4] C. C. Aggarwal, *Data mining: the textbook*. Springer, 2015.

[5] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, “A comparative evaluation of outlier detection algorithms: Experiments and analyses,” *Pattern Recognition*, vol. 74, pp. 406–421, 2018.

[6] Y. Zhao, Z. Nasrullah, and Z. Li, “PyOD: A python toolbox for scalable outlier detection,” *Journal of Machine Learning Research*, vol. 20, May 2019.

[7] K. Liu *et al.*, “PyGOD: A Python Library for Graph Outlier Detection,” *arxiv.org*, Accessed: May 01, 2022. [Online]. Available: https://arxiv.org/abs/2204.12095

[8] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: identifying density-based local outliers,” in *ACM sigmod record*, 2000, vol. 29, no. 2, pp. 93–104.

[9] “local outlier factor - wikipedia\_2022,” *En.wikipedia.org*. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Local\_outlier\_factor

[10] F. Angiulli and C. Pizzuti, “Fast outlier detection in high dimensional spaces,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2431 LNAI, pp. 15–27, 2002, doi: 10.1007/3-540-45681-3\_2.

[11] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient algorithms for mining outliers from large data sets,” in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 427–438.

[12] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[13] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation-based anomaly detection,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.

[14] L. Breiman, “Random forests,” *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001.

[15] J. Verbus, “Detecting and preventing abuse on LinkedIn using isolation forests,” *Engineering.linkedin.com*. 2022. [Online]. Available: https://engineering.linkedin.com/blog/2019/isolation-forest

[16] M. Goldstein, A. D.-K.-2012: poster and demo track, and undefined 2012, “Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm,” *Citeseer*, Accessed: May 01, 2022. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.401.5686&rep=rep1&type=pdf

[17] G. O. Campos *et al.*, “On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study,” *Data Min Knowl Discov*, vol. 30, no. 4, pp. 891–927, 2016.

[18] S. Rayana, “ODDS Library.” 2016. [Online]. Available: http://odds.cs.stonybrook.edu

[19] D. Dua and C. Graff, “UCI Machine Learning Repository.” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[20] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, “OpenML: networked science in machine learning,” *ACM SIGKDD Explorations Newsletter*, vol. 15, no. 2, pp. 49–60, 2014.

[21] J. Davis and M. Goadrich, “The relationship between Precision-Recall and ROC curves,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.