2016 IEEE International Conference on Data Science and Advanced Analytics

On the Evaluation of Outlier Detection and One-Class Classification Methods

Lorne Swersky*, Henrique O. Marques†, Jörg Sander*, Ricardo J. G. B. Campello† and Arthur Zimek‡
*Department of Computing Science, University of Alberta, Edmonton, Canada
{swersky, jsander}@ualberta.ca
†Department of Computer Sciences, University of São Paulo, São Carlos, Brazil
{hom, campello}@icmc.usp.br
‡Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark
zimek@imada.sdu.dk

***Abstract*—It has been shown that unsupervised outlier detection methods can be adapted to the one-class classification problem. In this paper, we focus on the comparison of one-class classification algorithms with such adapted unsupervised outlier detection methods, improving on previous comparison studies in several important aspects. We study a number of one-class classification and unsupervised outlier detection methods in a rigorous experimental setup, comparing them on a large number of datasets with different characteristics, using different performance measures. Our experiments led to conclusions that do not fully agree with those of previous work.**

***Keywords*-semi-supervised learning; unsupervised learning; one-class classification; outlier detection; machine learning algorithms; predictive models; evaluation**

I. Introduction

Outlier detection is one of the central tasks of data mining. This task is aimed to identify those observations which deviate substantially from the remaining data. Many definitions of outlier exist in the literature. One of the most used is Hawkins' definition [10], which refers to an outlier as "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism". Detecting such patterns is important because they might represent extraordinary behaviors that deserve some special attention, such as a traffic accidents and network intrusion attacks.

Outlier detection algorithms can be categorized in supervised, semi-supervised, and unsupervised techniques [9]. Supervised techniques can be seen as a special case of binary classification, where there are enough observations labeled as inliers and outliers available to train a classifier using approaches for imbalanced classification [1]. In semi-supervised outlier detection, due to the rarity of the outliers, there are only very few or even no outlier observations available to sufficiently describe the outlier class. In this scenario, also referred to as "novelty detection", the model is typically obtained using one-class classification techniques [22]. When no labeled data are available, unsupervised techniques can be used that do not assume any prior knowledge about which observations are outliers and which are inliers [24].

In this paper, we focus on the comparison of one-class classification methods with unsupervised outlier detection methods adapted to the problem of novelty detection. Following the approach proposed in [14], unsupervised outlier detection methods can be extended to use inlier class information to be applicable also in the semi-supervised setting.

Janssens *et al.* [13] performed a comparative study between 3 methods proposed for one-class classification (kNN Data Description [7], Parzen Windows [20] and SVDD [26]) and 2 methods originally proposed for unsupervised outlier detection (LOF [4] and LOCI [19]) extended to the one-class classification scenario. The authors concluded that LOF and SVDD are the top two performers and that they are not distinguishable from a statistical significance test perspective.

In this paper, we perform a more comprehensive investigation, using a more rigorous experimental setup, which leads to conclusions that do not fully agree with Janssens *et al.* [13]. In particular, we make the following contributions:

- When reproducing the experiments in [13], which reports averages over 5 repetitions of 5-fold cross-validation, we noticed a large variability in the results. In order to increase results confidence, we perform 30 repetitions using 10-fold cross-validation instead.
- We increase the number of datasets used in the evaluation from 24 to 433 (33 base datasets plus 400 dataset variants from an image collection), and the number of compared methods from 5 to 11.
- In addition to the well-known Area Under the ROC curve (ROC AUC), used in Janssens *et al.* [13], we also use Adjusted Precision-at-*n* (AjustedPrec@*n*), as defined in [6], to measure performance. These measures complement each other and together give a more complete picture of the performance characteristic of a method [6].
- In addition to the type of experiment performed by Janssens *et al.* [13], where one class is labeled as inlier and the other classes are relabeled outliers, we also conduct a second type of experiment where one class is labeled as outlier and the other classes are relabeled inliers. Both types represent possible real appli-

978-1-5090-5206-6/16 $31.00 © 2016 IEEE
DOI 10.1109/DSAA.2016.8

cation scenarios. We show that some of the conclusions change depending on the type of experiment.

- We also include an adaptation of a recent outlier detection method — called GLOSH [5] — to the one-class classification problem.
- Last but not least, we discuss basic principles of one-class classification that should not be violated when adapting unsupervised outlier detection methods to this task. We carefully examined and adjusted all the codes used in our experiments, making sure that these principles were not violated when producing the reported results.

The remainder of this paper is organized as follows. In Section II we discuss related work. In Section III we provide the reader with relevant background on one-class classification and unsupervised outlier detection. In Section IV we briefly review the methods compared in our experiments. We describe the setup of the experiments in Section V and discuss the results in Section VI. Finally, we summarize and conclude the paper in Section VII.

II. Related Work

In this section we discuss related work that compares one-class classification methods with unsupervised outlier detection methods.

Although both unsupervised outlier detection and one-class classification were first studied in the statistics field [2], [18], little has been done in the literature in order to compare the performance of both categories of algorithms. The one-class classification scenario generally is an easier task, because training data for one class is available. While in the one-class classification setting one can estimate the probability density function (p.d.f.) or other models without considering the presence of outliers in the dataset, in the unsupervised outlier detection setting one must deal with possible outliers while estimating the p.d.f. or other models. Given the differences between these two settings, the corresponding methods cannot be compared in a straightforward way.

One attempt to compare one-class classification and unsupervised outlier detection methods was done by Hido *et al.* [11]. In that work, the authors compared their proposed outlier detection algorithm against other approaches, including supervised, semi-supervised and unsupervised outlier detection techniques. The comparison, however, was not entirely fair since most compared algorithms had their parameters tuned using cross-validation, while only 3 different values for LOF's parameter were tested.

Janssens *et al.* [14] proposed a methodological framework to make unsupervised outlier detection algorithms work in a one-class classification setup and thus be able to compare them against algorithms specifically designed for this task. The authors, however, only assessed the performance of two unsupervised methods, namely, LOF and LOCI, in the one-class classification scenario. In a follow-up work [13], the same framework previously proposed in [14] was once again applied to LOF and LOCI, but this time these methods were also compared against three one-class classification methods. The authors concluded that LOF and SVDD are the top two performers with an identical average performance rank, although each method has particular scenarios where one may outperform the other.

III. Background

A. One-Class Classification

Unlike in the traditional classification problem, in one-class classification [27] we are only provided with observations from one class and our model must then classify new observations as belonging to this class or not.

In order to keep terminology consistent, we will refer to observations belonging to the provided class as inliers, and observations not belonging to this class as outliers.

One-class classifiers can be categorized into density methods, boundary methods, and reconstruction methods. Commonly used density methods for one-class classification are the Gaussian density [3], Mixture of Gaussians [3], and Parzen density estimation [20]. In density estimation, the parameters for some p.d.f. can be fit using the training data, and then new observations can be classified using this p.d.f. Since the p.d.f. is estimated using only inliers, there is no risk of outliers affecting the distribution. One drawback, however, is that we require a sufficiently large sample of inliers to produce a good estimation. Depending, *e.g.*, on the dimensionality of the problem, the number of observations required to sufficiently represent the underlying distribution can become very large, and may prove too expensive to obtain in a real-world setting.

Boundary methods avoid the requirement for a large number of samples by instead attempting to define a boundary around the training data, such that new observations that fall within the boundary are classified as inliers, while observations falling outside of the boundary are classified as outliers. Since we are only interested in defining this boundary, it is not necessary to obtain a large number of samples to fully represent the inlier class. Boundary methods include the Support Vector Data Description (SVDD) [26] and the Linear Programming Distance Data Description (LP) [21].

Lastly, reconstruction methods can be used to model the training data by using a generating process. Such a generating process is chosen to provide a compact representation of the data while attempting to preserving most of the information as well as filtering out noise. Once the model has been obtained, new observations can then be described through this model in order to be classified. Reconstruction methods include approaches like auto-encoder networks [15].

B. Unsupervised Outlier Detection

In unsupervised outlier detection, we are provided with a set of unlabeled observations and are tasked with determining whether each observation is an inlier or an outlier. In this situation, an outlier may be defined as an observation that deviates from other observations in some significant way, as determined by the chosen method. There are a variety of approaches, including statistical, density-based, and cluster-based methods.

Statistical outlier detection methods [10] assume that the inliers were generated using some known parametric type of probability density function (p.d.f.). Potential drawbacks of these methods in the unsupervised setting are (1) the data may now contain outliers that will influence the estimated parameters of the assumed density, and (2) we rarely know in advance the true distribution of the data.

Density-based methods assume that outliers will appear in a region of low density, according to some non-parametric measure of density. Density-based approaches generally fall into global and local density methods. With global density methods, the density around an observation is compared with some density measure for the entire dataset. If the observation's density is sufficiently low, it is considered an outlier. In contrast, local density methods compare the density of an observation to that of its neighbors, rather than the entire dataset. Density-based methods include LOF [4] and LOCI [19].

Cluster-based methods use a clustering of the data in order to detect outliers. The intuition behind these methods is that observations that do not fit well into clusters can be considered outliers. A recent example is GLOSH, which is based on the HDBSCAN* clustering hierarchy [5].

C. Adapting Unsupervised Outlier Detection Methods to the One-Class Classification Setting

Common to unsupervised outlier detection methods is that they compute a certain outlier score for each observation. To use an unsupervised outlier detection method in one-class classification, the general strategy is as follows: First run the unsupervised method on the (one-class) training data, pre-computing the scores for each inlier. Then compute the score for a new observation to be classified, possibly using other pre-computed quantities (*e.g.*, densities or distances to nearest neighbors) related to observations in the training data. Then, in order to classify the new observation, compare its score with the pre-computed scores for the inliers.

There are two important aspects related to the use of pre-computed quantities that involve training data when computing the outlier score for a new observation to be classified. First, when classifying multiple observations, there is no need to recompute these quantities over and over again for each new observation, since they relate solely to the training data (inlier class model) and can thus be pre-computed, which makes computations faster.

Second and more importantly, using pre-computed quantities regarding the observations in the training data assures that the model is by no means affected by new observations to be classified. This is a basic principle of one-class classification, i.e., unlabeled observations should not affect the pre-computed inlier class model, since they may be outliers. For example, suppose that a certain algorithm operates by comparing the density of a new observation to be classified against the densities of its nearest neighbors among the known inliers (training data). In this case, the densities of the inliers should be pre-computed, not to be affected by the presence of the unlabeled observation being currently assessed; otherwise, each unlabeled observation would affect the model in a different way, which means that different observations would be classified by different models/criteria.

When classifying multiple observations, it is also recommended that the classification procedure described above be performed independently for each observation. This way, different unlabeled observations will not affect each other's assessment. The reason we only classify one observation at a time instead of multiple observations at once is because we make no assumptions about the nature of each observation in relation to the combined dataset as a whole. It is possible that observations to be classified, while outliers in the sense that they do not belong to the inlier class, may be grouped together in such a way that an unsupervised method would not detect them as ouliers.

IV. Compared Methods

Support Vector Data Description

Support Vector Data Description (SVDD) [26] is a boundary-based one-class classification method inspired by Support Vector Machines (SVM) [28] used in regular classification problems. The primary difference between SVDD and SVM is that while SVM attempts to separate two or more classes with a maximum margin hyperplane, SVDD instead will enclose the inlier class in a minimum volume hypersphere by minimizing the following error:

$$\mathcal{E}(R, \mathbf{a}, \boldsymbol{\xi}) = R^2 + C \sum_i \xi_i \tag{1}$$

subject to the constraints:

$$\|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \quad \forall i \tag{2}$$

where R is the radius of the hypersphere, $\mathbf{a}$ is the center of the hypersphere, $\boldsymbol{\xi}$ are slack variables allowing training observations $\mathbf{x}$ to fall outside the SVDD boundary, and C is a penalty (regularization) parameter.

Like traditional SVMs, the above formulation can also be extended to non-linearly transformed spaces using kernel methods. In our experiments we use a Gaussian kernel.

Gaussian Data Description

In the Gaussian Data Description [27], the Gaussian probability density function:

$$p_{Gauss}(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\, \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})} \quad (3)$$

is fit to the inlier data, where $\boldsymbol{\mu}$ is the mean and Σ is the covariance matrix, and d is the dimensionality of the data. A new observation can be classified by computing its probability under the learned distribution.

Parzen Window Data Description

Parzen Window Data Description (PW) is based on Parzen Density Estimation [20] which estimates the density of the data using a mixture of kernels centered on each of the N individual training observations. In our case, we use Gaussian kernels with diagonal covariance matrices $\Sigma_i = \lambda I$, where λ is a width parameter which can be optimized using the maximum likelihood solution. The probability of an observation being an inlier is then computed as:

$$PW(\mathbf{x}) = \frac{1}{N}\sum_i p_{Gauss}(\mathbf{x}|\mathbf{x}_i, \lambda I) \quad (4)$$

Unlike other density methods, PW is non-parametric and therefore classifying new observations can be relatively expensive.

Linear Programming Distance Data Description

The Linear Programming Distance Data Description (LP) [21] is a boundary method which utilizes a dissimilarity measure to compare new observations to inlier observations in the training set, $\mathcal{D}_{\text{train}}$. The dissimilarity measure used must meet a number of criteria defined by the authors. One such measure they propose and which we use in our experiments is the sigmoid function. LP constructs a boundary by bringing a hyperplane which bounds the training set from above in the dissimilarity space as close to the origin as possible while still accepting most inliers.

Auto-Encoder Data Description

In the Auto-Encoder Data Description [27], a neural network with hyperbolic tangent sigmoid units, a single hidden layer, and a parameter-defined number of hidden units is trained on the inlier class. In order to classify new observations, each observation to be classified is supplied as input to the network, and the difference between the original input and the network's output in terms of mean squared error is computed.

The Auto-Encoder Data Description falls within the category of reconstruction methods for one-class classification.

k-Nearest Neighbor Data Description

k-Nearest Neighbor Data Description [7], [13], which we call here kNN$_{local}$, is similar to LOF and LOCI in that it approximates the local density of the training observations, however in a simpler way. An observation is classified under kNN$_{local}$ by computing the ratio between the distance from an observation to its k^{th} nearest neighbor $\mathrm{NN_k}(\mathbf{x}_i)$, and the distance between the k^{th} nearest neighbor and its k^{th} nearest neighbor:

$$\mathrm{kNN_{local}}(\mathbf{x}_i, k) = \frac{d(\mathbf{x}_i, \mathrm{NN_k}(\mathbf{x}_i))}{d(\mathrm{NN_k}(\mathbf{x}_i), \mathrm{NN_k}(\mathrm{NN_k}(\mathbf{x}_i)))} \quad (5)$$

k-Nearest Neighbor Outlier Detection

The k-nearest neighbor outlier detection method, which we call kNN$_{global}$, has been originally introduced as an unsupervised distance-based outlier detection method [23]. Its score is the numerator of Equation (5):

$$\mathrm{kNN_{global}}(\mathbf{x}_i, k) = d(\mathbf{x}_i, \mathrm{NN_k}(\mathbf{x}_i)) \quad (6)$$

This makes the score global rather than local.

Local Outlier Factor

Local Outlier Factor (LOF) [4] is an unsupervised outlier detection method which functions similarly to kNN$_{local}$ by comparing the local density of an observation to that of its neighbors. The distances between observations are replaced by reachability distances, defined as:

$$\text{reach-dist}_k(\mathbf{x}_i \leftarrow \mathbf{x}_j) = \max\{d(\mathbf{x}_j, \mathrm{NN}_k(\mathbf{x}_j)), d(\mathbf{x}_i, \mathbf{x}_j)\} \quad (7)$$

The local reachability density of an observation $\mathbf{x}_i$ is then defined as the inverse average reachability distance from the set of $\mathbf{x}_i$'s neighbors, $k\mathrm{NN}(\mathbf{x}_i)$, that are within the k nearest neighbor distance around $\mathbf{x}_i$:

$$\mathrm{lrd_k}(\mathbf{x}_i) = \frac{|\,k\mathrm{NN}(\mathbf{x}_i)|}{\sum_{\mathbf{x}_j \in k\mathrm{NN}(\mathbf{x}_i)} \text{reach-dist}_k(\mathbf{x}_i \leftarrow \mathbf{x}_j)} \quad (8)$$

Finally, the LOF score of an observation is computed by comparing the lrd of the observation with that of its neighbors:

$$\mathrm{LOF}_k(\mathbf{x}_i) = \frac{\sum\limits_{\mathbf{x}_j \in k\mathrm{NN}(\mathbf{x}_i)} \frac{\mathrm{lrd}_k(\mathbf{x}_j)}{\mathrm{lrd}_k(\mathbf{x}_i)}}{|\,k\mathrm{NN}(\mathbf{x}_i)|} \quad (9)$$

Local Correlation Integral

Local Correlation Integral (LOCI) [19] is an unsupervised outlier detection method which analyzes the density of an observation at multiple neighborhood radii αr of a given maximum radius r, where $\alpha \in (0, 1]$. For each observation $\mathbf{x}_i$, a (local) r-neighborhood $\mathcal{N}(\mathbf{x}_i, r) = \{\mathbf{x}|d(\mathbf{x}_i, \mathbf{x}) \leq r\}$ and a (local) r-density $n(\mathbf{x}_i, r) = |\mathcal{N}(\mathbf{x}_i, r)|$ are defined.

The average αr-density inside an r-neighborhood around an observation $\mathbf{x}_i$ is then defined as:

$$\hat{n}(\mathbf{x}_i, r, \alpha) = \frac{\sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i, r)} n(\mathbf{x}_j, \alpha r)}{n(\mathbf{x}_i, r)} \tag{10}$$

and the multi-granularity deviation factor (MDEF) is given by:

$$\text{MDEF}(\mathbf{x}_i, r, \alpha) = 1 - \frac{n(\mathbf{x}_i, \alpha r)}{\hat{n}(\mathbf{x}_i, r, \alpha)} \tag{11}$$

An observation $\mathbf{x}_i$ is classified using the following score:

$$\sigma\,\text{MDEF}(\mathbf{x}_i, r, \alpha) = \frac{\sigma_{\hat{n}}(\mathbf{x}_i, r, \alpha)}{\hat{n}(\mathbf{x}_i, r, \alpha)}, \tag{12}$$

which is the normalized standard deviation $\sigma_{\hat{n}}(\mathbf{x}_i, r, \alpha)$ of $n(\mathbf{x}_i, \alpha r)$ for $\mathbf{x}_i \in \mathcal{N}(\mathbf{x}_i, r)$. With these quantities, the LOCI score is computed as follows:

$$\text{LOCI}(\mathbf{x}_i, \alpha) = \max_{r \in \mathcal{R}} \left\{ \frac{\text{MDEF}(\mathbf{x}_i, r, \alpha)}{\sigma\,\text{MDEF}(\mathbf{x}_i, r, \alpha)} \right\} \tag{13}$$

Angle-Based Outlier Detection

The intuition behind Angle-Based Outlier Detection (ABOD) [16] is that by measuring the variance in the angles between an observation and pairs of other observations, we can determine whether or not an observation is an outlier. If the variance is high, it suggests that the observation is surrounded by other observations (in a cluster), while a low variance suggests that the observation is far away from other observations (an outlier). The Angle-Based Outlier Factor (ABOF) is defined as follows:

$$\text{ABOF}(\mathbf{x}_i) = \text{VAR}_{\mathbf{x}_j, \mathbf{x}_k \in \mathcal{D}_{\text{train}}} \left(\frac{\langle \mathbf{x}_i - \mathbf{x}_j, \mathbf{x}_i - \mathbf{x}_k \rangle}{\|\mathbf{x}_i - \mathbf{x}_j\|^2 \cdot \|\mathbf{x}_i - \mathbf{x}_k\|^2} \right) \tag{14}$$

Global-Local Outlier Scores from Hierarchies

The Global-Local Outlier Scores from Hierarchies (GLOSH) [5] is an unsupervised outlier detection algorithm based on the hierarchical density estimates provided by the hierarchical clustering algorithm HDBSCAN*. After a density-based clustering hierarchy is computed for the whole dataset, the GLOSH score for each observation $\mathbf{x}_i$ can be computed based on the difference in density around $\mathbf{x}_i$ and the highest density inside the cluster closest to $\mathbf{x}_i$ (from a density-connectivity perspective) in the HDBSCAN* hierarchy, as follows:

$$\text{GLOSH}(\mathbf{x}_i) = \frac{\lambda_{\max}(C_{\mathbf{x}_i}) - \lambda(\mathbf{x}_i)}{\lambda_{\max}(C_{\mathbf{x}_i})} \tag{15}$$

where $\lambda(\mathbf{x}_i)$ is the density of $\mathbf{x}_i$ and $\lambda_{\max}(C_{\mathbf{x}_i})$ is the highest density of an observation inside the closest cluster $C_{\mathbf{x}_i}$, where densities are estimated by a k-nearest neighbor density estimator. The closest cluster $C_{\mathbf{x}_i}$ is the one that $\mathbf{x}_i$ belongs to at the density level of $\mathbf{x}_i$.

To apply GLOSH in a one-class classification scenario, we can construct initially the HDBSCAN* hierarchy using the training data, and then use this hierarchy as a fixed "model" to compute outlier scores for unseen data. This can be achieved by first adding a given observation $\mathbf{x}_i$ to the Minimum Spanning Tree (MST) which underlies the HDBSCAN* hierarchy; $\mathbf{x}_i$ is connected to the training observation $\mathbf{x}$ with the smallest "distance" in the density space in which the MST is constructed. Then, the cluster $C_{\mathbf{x}_i}$ that is closest to $\mathbf{x}_i$ in the hierarchy is determined as the cluster that is closest to $\mathbf{x}$.

V. Experimental Setup

We compare and evaluate the 11 algorithms described in the previous section: ABOD, Auto-Encoder, Gaussian Density, GLOSH, kNN$_{global}$, kNN$_{local}$, LOCI, LOF, Linear Programming, Parzen Windows and SVDD. We use code from the repository available at http://prlab.tudelft.nl/users/david-tax/ [25] for most compared algorithms, except for LOF, LOCI, kNN$_{local}$ and GLOSH. In the case of LOF and LOCI, their implementations were modified to ensure that new observations to be classified do not affect the pre-computed model for the inlier class, following the guidelines previously discussed in Section III-C. As KNN$_{local}$ was not available in that repository, we used our own implementation for this algorithm. GLOSH was adapted based on the implementation of HDBSCAN* made available at http://lapad-web.icmc.usp.br/.

We use 31 real-world datasets from the UCI Machine Learning Repository [17] as pre-processed for one-class classification and made available at http://prlab.tudelft.nl/users/david-tax/: Abalone, Arrhythmia, Balance-scale, Ball-bearing, Biomed, Breast, Cancer, Colon, Delft1x3, Delft2x2, Delft3x2, Delft5x1, Delft5x3, Diabetes, Ecoli, Glass, Heart, Hepatitis, Housing, Imports, Ionosphere, Iris, Liver, Satellite, Sonar, Spectf, Survival, Vehicle, Vowels, Waveform and Wine.

In addition, we use CellCycle-237 and YeastGalactose, made public by Yeung *et al.* [29], [30], as well as a collection of 400 datasets based on the Amsterdam Library of Object Images (ALOI) [8], created as described in [12]. Specifically, this collection has been created by randomly selecting 2, 3, 4 or 5 ALOI image categories as class labels and then sampling 25 images from each of the selected categories, thus resulting in datasets containing 2, 3, 4, or 5 classes and 50, 75, 100, or 125 images (observations). Following [12], these images are described by six descriptors: color moments (144 attributes), texture statistics extracted from the gray-level co-occurrence matrix (88 attributes), Sobel edge histogram (128 attributes), first-order statistics from the gray level histogram (5 attributes), gray-level run-length matrix features (44 attributes), and gray-level histogram (256 attributes). PCA is then applied to each set of attribute vectors separately and the first principal component resulting from each set is extracted. The extracted first components are then combined in such a way that each image is thus described by a vector with six attributes.

In total, we have 433 *real-world* multi-class datasets. We average the results for the ALOI and Delft datasets, for they are variants obtained from the same source. Finally, due the inability of some algorithms to deal with replicated observations, duplicates are removed from the datasets where they are present.

In order to evaluate a method's performance on a one-class dataset, we perform the following procedure: First, we split the dataset into 2 subsets, one containing 20% and the other containing 80% of the data. In the subset with 80% of the data we apply a 10-fold cross-validation procedure to optimize the parameters of the methods with respect to ROC AUC.

The parameters of the methods were optimized in the following ranges: $k = 1, 2, \cdots, 50$ for LOF, kNN$_{global}$ and kNN$_{local}$; $M_{clSize} = M_{pts} = 1, 2, \cdots, 50$ for GLOSH; No. hidden units = 2, 5, 7, 10, 12, 15, 17, 20, 22, 25 for Auto-Encoder[1], $h = 0.001$ to 50 (discretized logarithmically in 25 different values) for Gaussian Density and for the Gaussian kernel used in SVDD, $\alpha = 0.1, 0.2, \cdots, 1.0$ for LOCI, LP, Parzen Windows and SVDD.

After parameter optimization, the subset containing 20% of the data (test set) is used to measure the performances of the methods (trained with the optimal parameter values from the 10-fold cross-validation). In order to get more reliable results, this procedure is repeated 30 times, and the resulting ROC AUC values are aggregated and reported.

For the sake of comparison with the results reported by Janssens *et al.*, we also compute the Weighted ROC AUC measure used in their work [13], which gives more weight to results obtained from experiments involving larger inlier classes. It is worth remarking, however, that this approach may be questionable, since it is well known that ROC curves already inherently adjust for the imbalance of class sizes.

In addition to ROC AUC values, we also report the adjusted precision-at-*n* measure (AjustedPrec@*n*) [6] for the classification of results obtained on the test sets. While ROC AUC takes the entire test set into account, precision-at-*n* (fraction of true outliers among the top n outlier scores in the test set) evaluates only the top *n* observations, where *n* is the number of observations known to be outliers in the test set. To compare precision-at-*n* values across datasets with different numbers of outliers, one has to adjust precision-at-*n* by chance, as discussed in [6], which gives rise to the AjustedPrec@n measure used in our experiments.

A high ROC AUC score indicates only that, in the overall outlier ranking, outliers are more likely to be ranked ahead of inliers; it does not necessarily mean that the top positions in the ranking are dominated by outliers. Therefore, following the extensive study in [6], we argue that one cannot rely solely on ROC AUC scores in judging the quality of an outlier method; rather, ROC AUC and AdjustedPrec@n complement each other, as they reveal different aspects of an outlier ranking, both of which are relevant in practice.

We perform two major types of experiments. In the **first type** of experiment (Type I), we follow the only approach taken by Janssens et al. [13], where multi-class datasets are transformed into one-class datasets by re-labeling one class as inliers, and the remaining classes as outliers. Except for datasets where only a single inlier class has been pre-defined as such in the data repository (http://prlab.tudelft.nl/users/david-tax/) — *e.g.*, Ecoli — we repeat the procedure for every class as inliers in a dataset, and average the results.

For the **second type** of experiment (Type II), we reverse the inlier and outlier classes obtained in the first type of experiment for datasets that have more than 2 possible inlier classes defined. This type of experiment is important since it models situations with a possibly multi-modal inlier class. Note that Type II experiments are only different from Type I — and are therefore only reported — for datasets with 3 or more classes. For this reason, results for Type II experiments are only available for a subset of the datasets considered in Type I experiments.

VI. Results

Figure 1 shows the average ranking of the 5 methods compared in [13] over all Type I experiments w.r.t. weighted ROC AUC. This figure summarizes our attempt at reproducing Janssens *et al.*'s results [13], which were restricted to this particular setup only. The width of the upper bar (CD) indicates the critical distance of the well-known Friedman/Nemenyi statistical test at significance level $\alpha = 0.05$. The analogous figure in [13] shows two subsets of methods: (1) the top performers SVDD, LOF, and KNN$_{local}$ (with SVDD and LOF having exactly the same average rank), and (2) a group consisting of PW and LOCI, clearly separated from (1), with much lower performance. Our result does not agree in several respects: first, LOF and SVDD are not tied, second, there are no longer two clearly separated groups, and third, KNN$_{local}$ is the worst performer rather than one of the best.

In the following, we describe the results according to our experimental setup described above. Detailed numbers for all experiments are given in tables in the Appendix. The overall ranking results are summarized in Figures 2 and 3.

Figure 2 shows the average rankings of the methods over all Type I experiments with respect to ROC AUC and AdjustedPrec@n. When looking at ROC AUC, one can see SVDD, Gaussian, and kNN$_{global}$, in this order,

[1] Due to the high computational demand of Auto-Encoder, a time limit of 1000s was imposed to the network convergence, which gives a maximum amount of time equal to 30 repetitions × 10 folds × 10 parameters × 1000s ≈ 35 days per experiment involving a single dataset.

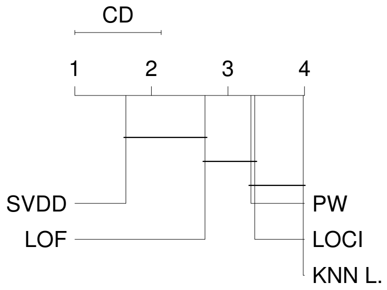
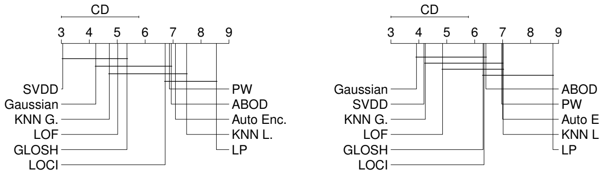


Figure 1: Average ranking of the methods compared in [13] over all Type I experiments w.r.t. weighted ROC AUC.



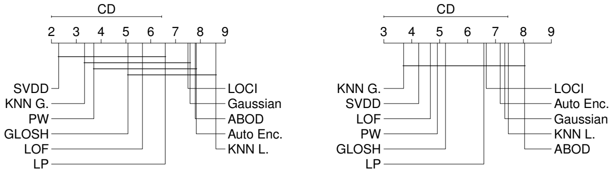
(a) ROC AUC

(b) AdjustedPrec@n

Figure 2: Average rankings of the methods over all Type I experiments.

at the top. The average ROC AUC for SVDD, Gaussian, and kNN$_{global}$ were 0.8, 0.8, and 0.78, respectively. When looking at AdjustedPrec@n the overall picture is similar, but AdjustedPrec@n tells us that SVDD is no longer the top performer as it is now outperformed by Gaussian. This suggests that in the outlier scoring produced by Gaussian there are on average more true outliers with top scores than in SVDD, whereas for SVDD the scores of the true outliers tend to be higher than those of inliers, overall.

Figure 3 shows the average rankings of the methods over all Type II experiments with respect to ROC AUC and AdjustedPrec@n. When comparing ROC AUC with AdjustedPrec@n, we can notice again some inversions in the ranks, *e.g.*, between LOF and PW/GLOSH. In particular, the relative performance of SVDD drops once more for AdjustedPrec@n, as it also does in Type I experiments, but now SVDD is outperformed by KNN$_{global}$, not by Gaussian. In fact, when comparing Type I experiments in Figure 2 and Type II experiments in Figure 3, a noticeable difference can be observed with respect to Gaussian: while it was among the top 3 performers in Type I experiments, its relative performance drops sharply in Type II. The absolute performance of Gaussian indeed drops from Type I to Type II experiments, from 0.8 to 0.77 for ROC AUC and from 0.48 to 0.38 for AdjustedPrec@n, which is expected as Gaussian presumes a unimodal inlier class model that fits best the data as arranged in Type I experiments. But this alone does not fully explain the drop of Gaussian in terms of relative performance. What also explains it is that other methods, particularly local density-based methods like GLOSH and LOF, perform better in Type II experiments (see the tables in the Appendix for detailed values), which correspond to application scenarios with possibly multi-modal target classes.



(a) ROC AUC

(b) AdjustedPrec@n

Figure 3: Average rankings of the methods over all Type II experiments.

Overall, based on both types of experiments, we conclude that: (i) in agreement with [13], SVDD is a top performer, especially with respect to ROC AUC; (ii) kNN$_{global}$, however, might be a preferable choice, since it is consistently a top performer, yet it is much simpler than SVDD; this method was not included in the study in [13]; and (iii) in contrast to [13], LOF does not perform as strongly as SVDD, and kNN$_{local}$ is not among the top performers, but consistently among the worst.

VII. Conclusion

In this paper we provided a comprehensive comparison of one-class classification algorithms and unsupervised outlier detection methods extended to the one-class classification scenario. These methods were evaluated over a number of datasets, measuring performance with respect to ROC AUC and AdjustedPrec@n in different experimental settings. The most important conclusion is that SVDD and kNN$_{global}$ are the top choices for one-class classification, while we do not recommend kNN$_{local}$. This is in contrast to the previous comparison study by Janssens *et al.* [13], which did not include kNN$_{global}$ and reported kNN$_{local}$ as a top performer. In addition, we could not confirm the top performance of LOF reported in [13], but only that of SVDD.

As additional contributions, we proposed and described an adaptation of a recent outlier detection method — called GLOSH [5] — to the one-class classification problem. We also discussed basic principles of one-class classification that should not be violated when adapting unsupervised outlier detection methods to this task, and how to assure that such principles are not violated.

Acknowledgment

This research was supported by the Research Foundation of the State of São Paulo — FAPESP (grant No. 2015/06019-0), the Brazilian National Research Council — CNPq (grants No. 304137/2013-8 & No. 400772/2014-0), and NSERC Canada.

References

[1] C. C. Aggarwal, *Outlier Analysis*. Springer, 2013.

[2] V. Barnett and T. Lewis, *Outliers in Statistical Data*, 3rd ed. John Wiley & Sons, 1994.

[3] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '00. New York, NY, USA: ACM, 2000, pp. 93–104.

[5] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 1, pp. 5:1–5:51, Jul. 2015. [Online]. Available: http://doi.acm.org/10.1145/2733381

[6] G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle, "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 891–927, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10618-015-0444-8

[7] D. de Ridder, D. M. Tax, and R. P. W. Duin, "An experimental comparison of one-class classification methods," in *Proc. 4th Annual Conference of the Advanced School for Computing and Imaging (ASCI'98)*, B. Ter Haar Romeny, D. Epema, J. Tonino, and A. Wolters, Eds., ASCI. Delft, The Netherlands: ASCI, 1998, pp. 213–218.

[8] J.-M. Geusebroek, G. J. Burghouts, and A. W. Smeulders, "The amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000042993.50813.60

[9] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2011.

[10] D. Hawkins, *Identification of Outliers*. Chapman and Hall, 1980.

[11] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Inlier-based outlier detection via direct density ratio estimation," in *2008 Eighth IEEE International Conference on Data Mining*, Dec 2008, pp. 223–232.

[12] D. Horta and R. J. G. B. Campello, "Automatic aspect discrimination in data clustering," *Pattern Recognition*, vol. 45, no. 12, pp. 4370–4388, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320312002415

[13] J. H. M. Janssens, I. Flesch, and E. O. Postma, "Outlier detection with one-class classifiers from ml and kdd," in *Machine Learning and Applications, 2009. ICMLA '09. International Conference on*, Dec 2009, pp. 147–153.

[14] J. H. M. Janssens and E. O. Postma, "One-class classification with LOF and LOCI: An empirical comparison," in *Proceedings of the 18th Annual Belgian-Dutch on Machine Learning*, May 2009, pp. 56–64.

[15] N. Japkowicz, C. Myers, M. Gluck *et al.*, "A novelty detection approach to classification," in *IJCAI*, 1995, pp. 518–523.

[16] H.-P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 444–452.

[17] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[18] M. Markou and S. Singh, "Novelty detection: a review–part 1: statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168403002020

[19] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos, "LOCI: fast outlier detection using the local correlation integral," Bangalore, India, 2003, pp. 315–326.

[20] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[21] E. Pekalska, D. M. J. Tax, and R. P. W. Duin, "One-class LP classifier for dissimilarity representations," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15. MIT Press: Cambridge, MA, 2003.

[22] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016516841300515X

[23] 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*, ser. SIGMOD '00. New York, NY, USA: ACM, 2000, pp. 427–438.

[24] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison Wesley, 2006.

[25] D. M. J. Tax, "DDtools, the data description toolbox for Matlab," June 2015, version 2.1.2.

[26] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[27] D. M. J. Tax, *One-class classification*. TU Delft, Delft University of Technology, 2001.

[28] V. N. Vladimir and V. Vapnik, "The nature of statistical learning theory," 1995.

[29] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, vol. 17, no. 10, pp. 977–987, 2001. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/17/10/977.abstract

[30] K. Y. Yeung, M. Medvedovic, and R. E. Bumgarner, "Clustering gene-expression data with repeated measurements," *Genome Biology*, vol. 4, no. 5, pp. 1–17, 2003. [Online]. Available: http://dx.doi.org/10.1186/gb-2003-4-5-r34

Appendix

Tables I and II display ROC AUC values for Type I and Type II experiments, respectively, for each method. Tables III and IV show the AjustedPrec@n values for Type I and Type II experiments, respectively, for each method. The highest achieved values for each data set are shown in bold.

Table I: First type of experiments — ROC AUC

ROC AUC	GLOSH	KNN L.	LP	ABOD	Auto Enc.	Gaussian	KNN G.	LOCI	LOF	PW	SVDD
Abalone	0.66	0.66	0.71	0.71	0.72	0.74	0.73	0.76	0.66	0.74	**0.77**
Aloi	0.98	0.97	0.95	**0.99**	0.98	**0.99**	**0.99**	0.98	0.98	0.98	0.98
Arrhythmia	0.63	0.58	0.5	0.51	0.53	0.55	0.52	0.53	0.6	0.5	**0.64**
Balance-Scale	0.88	0.86	0.88	0.86	0.91	**0.93**	0.87	0.87	0.92	0.87	0.91
Ball-Bearing	0.98	0.97	0.5	0.93	**1**	**1**	0.98	0.96	0.99	0.53	0.98
Biomed	**0.84**	0.81	0.51	0.65	0.72	0.8	**0.84**	0.79	0.71	0.69	0.7
Breast	0.96	0.93	0.81	0.93	0.91	**0.98**	0.96	**0.98**	0.96	0.8	**0.98**
Cancer	0.52	0.52	0.5	0.53	0.54	**0.59**	0.54	0.53	0.53	0.51	0.53
CellCycle237	0.81	0.72	0.74	0.81	0.76	0.82	**0.84**	0.72	0.81	0.74	0.83
Colon	0.67	0.63	0.5	0.64	0.58	0.67	0.66	0.59	**0.68**	0.5	**0.68**
Delft	0.95	**0.96**	0.93	0.68	0.83	0.95	0.93	0.89	**0.96**	0.93	**0.96**
Diabetes	0.65	0.63	0.51	0.61	0.62	0.64	0.62	0.63	**0.66**	0.59	0.65
Ecoli	**0.94**	0.93	0.92	0.93	0.89	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**
Glass	0.78	0.78	0.81	0.79	0.76	0.78	0.81	0.67	0.81	0.81	**0.82**
Heart	0.6	0.57	0.5	0.59	0.67	**0.73**	0.58	0.58	0.59	0.55	0.61
Hepatitis	0.56	0.53	0.5	0.56	0.73	**0.74**	0.56	0.55	0.54	0.56	0.57
Housing	0.65	0.65	0.58	0.68	0.76	**0.78**	0.69	0.66	0.65	0.7	0.7
Imports	0.7	0.68	0.81	0.66	0.69	0.65	0.71	0.73	0.78	**0.83**	0.74
Ionosphere	**0.74**	0.66	0.66	0.64	0.62	0.64	0.67	0.6	0.64	0.64	**0.74**
Iris	0.97	0.95	**0.98**	0.97	0.96	**0.98**	0.97	0.97	0.97	**0.98**	**0.98**
Liver	0.54	0.55	0.53	0.55	0.54	0.54	0.55	**0.58**	0.55	0.53	0.56
Satellite	0.95	0.92	0.5	0.95	0.9	0.94	**0.96**	0.94	0.93	0.92	**0.96**
Sonar	0.7	0.73	0.76	0.64	0.67	0.66	0.76	0.65	**0.77**	0.76	0.75
Spectf	**0.66**	0.61	0.5	0.56	0.57	0.55	0.52	0.61	0.63	0.64	**0.66**
Survival	0.61	0.58	0.56	0.56	0.57	0.58	0.62	0.62	0.61	0.55	**0.65**
Vehicle	0.75	0.77	0.5	0.76	0.83	**0.9**	0.79	0.76	0.77	0.79	0.82
Vowels	0.99	0.98	**1**	0.99	0.63	0.99	**1**	0.91	0.99	**1**	0.99
Waveform	0.89	0.85	0.87	0.9	0.86	**0.91**	0.89	0.89	0.88	0.88	**0.91**
Wine	0.86	0.85	0.57	0.88	0.85	**0.96**	0.86	0.86	0.86	0.83	0.87
YeastGalactose	**0.99**	**0.99**	0.98	**0.99**	0.97	0.98	**0.99**	0.75	**0.99**	**0.99**	**0.99**
	0.78	0.76	0.69	0.75	0.75	0.8	0.78	0.75	0.78	0.74	0.8

Table II: Second type of experiments — ROC AUC

ROC AUC	GLOSH	KNN L.	LP	ABOD	Auto Enc.	Gaussian	KNN G.	LOCI	LOF	PW	SVDD
Abalone	0.62	0.62	0.7	0.67	0.65	0.7	0.68	0.67	0.63	0.71	**0.73**
Aloi	**0.96**	0.94	0.95	0.92	0.94	0.92	**0.96**	0.92	**0.96**	**0.96**	**0.96**
Balance-Scale	0.82	0.8	0.79	0.79	0.77	0.78	0.83	0.81	**0.84**	0.82	0.81
CellCycle237	**0.81**	0.69	0.73	0.75	0.76	0.76	0.79	0.71	0.75	0.74	0.78
Glass	0.75	0.71	0.76	0.69	0.69	0.64	0.76	0.7	**0.77**	0.76	**0.77**
Iris	0.95	0.93	**0.97**	0.96	0.94	0.82	0.96	0.95	0.93	**0.97**	**0.97**
Satellite	**0.85**	0.79	0.5	0.74	0.77	0.74	0.84	0.77	0.82	0.82	**0.85**
Vehicle	0.68	0.63	0.5	0.61	0.66	0.71	0.74	0.7	0.67	0.72	**0.75**
Vowels	0.94	0.94	**0.98**	0.75	0.7	0.64	**0.98**	0.09	0.95	**0.98**	**0.98**
Waveform	0.81	0.69	0.75	0.75	0.62	0.76	0.81	0.83	0.75	0.77	**0.86**
Wine	0.74	0.73	0.58	0.76	0.84	**0.85**	0.75	0.71	0.74	0.76	0.77
YeastGalactose	0.95	0.91	**0.97**	0.93	0.93	0.91	**0.97**	0.93	0.95	**0.97**	**0.97**
	0.82	0.78	0.76	0.78	0.77	0.77	0.84	0.73	0.81	0.83	0.85

Table III: First type of experiments — AdjustedPrec@n

AdjustedPrec@n	GLOSH	KNN L.	LP	ABOD	Auto Enc.	Gaussian	KNN G.	LOCI	LOF	PW	SVDD
Abalone	0.21	0.17	0.3	0.32	0.29	0.32	0.33	0.37	0.2	0.31	**0.4**
Aloi	0.92	0.87	0.85	0.93	0.92	**0.95**	0.94	0.91	0.91	0.92	0.92
Arrhythmia	-0.19	0.14	-0.77	0.03	0.08	0.09	0.05	0.05	**0.17**	-0.77	-0.19
Balance-Scale	0.44	0.5	0.53	0.51	0.61	0.59	0.44	0.55	0.61	0.56	**0.64**
Ball-Bearing	0.84	0.78	-0.28	0.7	0.97	**0.98**	0.85	0.76	0.88	-0.2	0.84
Biomed	**0.57**	0.52	-0.54	0.31	0.4	0.54	**0.57**	0.45	0.37	0.06	0.08
Breast	0.84	0.75	0.43	0.75	0.72	0.87	0.84	**0.88**	0.83	0.38	0.87
Cancer	0.05	0.06	-0.32	0.06	0.03	0.08	**0.11**	-0.01	0.08	-0.32	-0.04
CellCycle237	0.41	0.32	0.4	0.45	0.42	0.51	0.5	0.34	0.44	0.45	**0.52**
Colon	0.2	0.16	-0.62	0.18	0.13	0.21	0.21	-0.17	0.2	-0.62	**0.27**
Delft	0.73	0.73	0.67	0.29	0.48	0.71	0.67	0.63	**0.77**	0.67	0.73
Diabetes	0.22	0.19	-0.52	0.18	0.16	0.22	0.18	0.21	**0.24**	0.11	0.23
Ecoli	**0.75**	0.7	0.66	0.72	0.59	0.73	**0.75**	0.72	0.74	0.74	0.74
Glass	0.4	0.38	0.47	0.45	0.37	0.41	0.48	0.24	0.46	**0.49**	**0.49**
Heart	0.16	0.11	-0.86	0.13	0.27	**0.34**	0.1	0.13	0.13	-0.23	0.17
Hepatitis	0.01	0.03	-0.28	0.04	0.24	**0.25**	0.01	0.02	0	-0.26	-0.02
Housing	0.11	0.13	0	0.14	0.2	**0.22**	0.18	0.16	0.13	0.14	0.16
Imports	0.35	0.28	0.45	0.21	0.31	0.22	0.36	0.38	0.41	**0.53**	0.32
Ionosphere	0.12	0.3	0.28	0.29	0.25	0.32	**0.39**	0.14	0.32	0.31	0.17
Iris	0.81	0.78	**0.88**	0.85	0.8	**0.88**	0.82	0.83	0.84	0.86	0.86
Liver	0.01	0.08	-0.62	0.06	0.05	0.06	0.05	**0.13**	0.07	-0.32	0.03
Satellite	0.73	0.59	-0.21	0.73	0.55	0.71	0.75	0.71	0.67	0.71	**0.77**
Sonar	0.26	0.34	0.38	0.18	0.23	0.21	0.39	0.23	**0.41**	0.39	0.3
Spectf	0.04	0.08	-0.26	0.06	0.05	0.08	0.07	**0.12**	0.11	0.04	0.05
Survival	0.15	0.12	0.04	0.1	0.11	0.13	0.19	0.17	0.17	0.06	**0.25**
Vehicle	0.35	0.37	-0.33	0.35	0.46	**0.62**	0.44	0.35	0.38	0.43	0.47
Vowels	0.87	0.83	**0.94**	0.83	0.84	0.82	**0.94**	0.82	0.86	**0.94**	**0.94**
Waveform	0.56	0.47	0.51	0.61	0.5	0.61	0.56	0.61	0.53	0.53	**0.62**
Wine	0.5	0.52	-0.28	0.56	0.49	**0.8**	0.51	0.52	0.51	0.47	0.53
YeastGalactose	**0.96**	0.9	0.88	0.94	0.88	0.9	**0.96**	0.44	0.91	0.92	0.95
	0.41	0.41	0.09	0.4	0.41	0.48	0.45	0.39	0.44	0.28	0.44

Table IV: Second type of experiments — AdjustedPrec@n

AdjustedPrec@n	GLOSH	KNN L.	LP	ABOD	Auto Enc.	Gaussian	KNN G.	LOCI	LOF	PW	SVDD
Abalone	0.12	0.2	0.22	0.21	0.18	**0.3**	0.23	0.26	0.21	**0.3**	0.16
Aloi	**0.81**	0.74	0.67	0.68	0.77	0.74	**0.81**	0.69	0.8	0.78	**0.81**
Balance-Scale	0.43	0.48	0.54	0.43	0.45	0.44	0.43	0.52	**0.59**	0.54	0.56
CellCycle237	**0.31**	0.18	0.04	0.22	0.24	0.23	0.28	0.17	0.3	0.21	0.27
Glass	0.29	0.28	**0.34**	0.2	0.22	0.13	0.32	0.21	**0.34**	**0.34**	0.17
Iris	0.8	0.75	**0.86**	0.83	0.78	0.58	0.82	0.79	0.76	0.85	**0.86**
Satellite	**0.47**	0.44	-0.21	0.21	0.35	0.31	0.46	0.29	0.45	0.32	0.44
Vehicle	0.25	0.17	-0.33	0.14	0.2	0.25	**0.28**	0.23	0.23	0.16	**0.28**
Vowels	0.58	0.58	**0.77**	0.14	0.19	0.08	**0.77**	0.3	0.61	**0.77**	0.76
Waveform	0.45	0.29	0.38	0.35	0.14	0.34	0.45	0.48	0.36	0.4	**0.53**
Wine	0.33	0.38	-0.51	0.37	0.54	**0.56**	0.37	0.37	0.38	0.26	0.31
YeastGalactose	0.81	0.64	0.82	0.74	0.69	0.64	**0.85**	0.73	0.73	0.84	**0.85**
	0.47	0.43	0.3	0.38	0.4	0.38	0.51	0.42	0.48	0.48	0.5