

# Unsupervised Parameter-free Outlier Detection using HDBSCAN\* Outlier Profiles

Kushankur Ghosh  
Department of Computing Science  
University of Alberta  
Edmonton, Canada  
kushanku@ualberta.ca

Murilo Coelho Naldi  
Department of Computing Science  
Federal University of São Carlos  
São Paulo, Brazil  
naldi@ufscar.br

Jörg Sander  
Department of Computing Science  
University of Alberta  
Edmonton, Canada  
jsander@ualberta.ca

Euijin Choo  
Department of Computing Science  
University of Alberta  
Edmonton, Canada  
euijin@ualberta.ca

**Abstract**—In machine learning and data mining, outliers are data points that significantly differ from the dataset and often introduce irrelevant information that can induce bias in its statistics and models. Therefore, unsupervised methods are crucial to detect outliers if there is limited or no information about them. *Global-Local Outlier Scores based on Hierarchies* (GLOSH) is an unsupervised outlier detection method within HDBSCAN\*, a state-of-the-art hierarchical clustering method. GLOSH estimates outlier scores for each data point by comparing its density to the highest density of the region they reside in the HDBSCAN\* hierarchy. GLOSH may be sensitive to HDBSCAN\*'s  $min_{pts}$  parameter that influences density estimation. With limited knowledge about the data, choosing an appropriate  $min_{pts}$  value beforehand is challenging as one or some  $min_{pts}$  values may better represent the underlying cluster structure than others. Additionally, in the process of searching for “potential outliers”, one has to define the number of outliers  $n$  a dataset has, which may be impractical and is often unknown. In this paper, we propose an unsupervised strategy to find the “best”  $min_{pts}$  value, leveraging the range of GLOSH scores across  $min_{pts}$  values to identify the value for which GLOSH scores can best identify outliers from the rest of the dataset. Moreover, we propose an unsupervised strategy to estimate a threshold for classifying points into inliers and (potential) outliers without the need to pre-define any value. Our experiments show that our strategies can automatically find the  $min_{pts}$  value and threshold that yield the best or near best outlier detection results using GLOSH.

**Index Terms**—Outlier Detection, Clustering, Parameter Estimation, HDBSCAN\*

## I. INTRODUCTION

An outlier is an observation that deviates significantly from other observations as to arouse suspicions that it was probably generated by a different mechanism [1]. Outliers are common in real-world datasets and often do not contribute to the discovery of knowledge about the data, degrading the performance of machine learning models. Sometimes, outliers can exhibit unique behaviors that eventually give insights into areas such as intrusion detection [2]. Hence, detecting outliers is important in machine learning and data mining.

Outlier detection finds deviating instances using supervised, unsupervised, or one-class classification (OCC) strategies. In a supervised setting, there are enough labeled examples of outliers and inliers to train a binary classifier (typically with an imbalance between the number of outliers and the number of inlier examples) [3]. In an OCC setting, we have information about inliers and little or no information about outliers [4], and a model is learned solely based on information about inliers. In an unsupervised setting, there is no prior knowledge about the data, and unsupervised methods identify outliers as instances that deviate from a reference set of data.

**Motivation.** *Global-Local Outlier Scores based on Hierarchies* (GLOSH) is an unsupervised outlier detection method which is a part of the HDBSCAN\* clustering framework [5]. It can detect the data points that deviate from their local neighborhood (so-called *local* outliers) and also the data points that differ more globally from the rest of the data (so-called *global* outliers). The GLOSH score of a data point  $p$  is computed as the normalized difference between the density estimated around  $p$  and the highest density estimated in the cluster closest to  $p$  in the HDBSCAN\* hierarchy. The hierarchy represents clusters formed at different density levels, and it is constructed w.r.t. a parameter value  $min_{pts}$ , a smoothing factor set by a user, that determines the density estimates and, thereby, the GLOSH score. The GLOSH score quantifies the outlieriness of a data point w.r.t. its closest cluster. However, different  $min_{pts}$  values can yield different HDBSCAN\* hierarchies, and one of them may better represent the intrinsic cluster structure of the data than others. An unsuitable  $min_{pts}$  value can result in outliers getting a low GLOSH score. It has been shown that one may need to use multiple values of  $min_{pts}$  to reveal all the clusters in a dataset [6]. This means that the closest clusters of a data point could be different depending on the value of  $min_{pts}$ , and hence a variety of outlier detection results are possible for different  $min_{pts}$  values. In practice, the underlying data distribution is unknown. Therefore, choosing a single  $min_{pts}$  value that

allows the detection of most (preferably all) “true outliers” from “true inliers” may be challenging.

Moreover, it is essential to pre-define a number  $n$  of data points with the highest GLOSH scores. These  $n$  data points are treated as “potential outliers” and are subsequently presented to the domain experts for labeling the “true outliers” [7]. In practice, the number of outliers in a dataset is unknown, which makes it challenging to pick a suitable value for  $n$  beforehand. **Contribution.** In this paper, we tackle the challenge of choosing a  $min_{pts}$  and  $n$  value that can potentially yield the best results for GLOSH. Considering multiple results with increasing  $min_{pts}$  values, we show that the GLOSH scores of the data points change at a similar rate at a particular  $min_{pts}$  value, resulting in the best overall performance. Based on this observation, we designed a strategy based on GLOSH to find the  $min_{pts}$  value that yields the best or nearly the best results in a given range of  $min_{pts}$  values. At this  $min_{pts}$  value, we observed that the distribution of GLOSH scores shows a pattern that we used as a key to develop a strategy that automatically finds a threshold to separate potential outliers from inliers. Our strategies can be used in a fully unsupervised way and computed efficiently by running HDBSCAN\* following [8]. In [8], it has been shown that extracting multiple hierarchies for a range of  $min_{pts}$  values is possible with a run-time comparable to running HDBSCAN\* twice. We summarize our major contributions as follows:

- We introduce the notion of a GLOSH–Profile of a data point  $p$  as a sequence of GLOSH scores for  $p$  with respect to a range of  $min_{pts}$  values (IV-A).
- We empirically show that scores in a GLOSH–Profile have a pattern that allows the identification of the  $min_{pts}$  value that best distinguishes inliers and outliers (IV-B and IV-C).
- We introduce automatic parameter selection using GLOSH, called Auto-GLOSH, a strategy leveraging GLOSH–Profiles to find the  $min_{pts}$  value that leads to the best outlier detection using GLOSH (IV-D).
- We introduce Potential Outlier Labelling AppRoach (POLAR), a strategy leveraging the distribution of GLOSH scores at the  $min_{pts}$  value estimated by Auto-GLOSH to find the threshold for labeling potential outliers (V-B).

## II. RELATED WORK

Unsupervised outlier detection is based on the premise that potential outliers lie in low-density regions of the reference set. In [9], outliers are data points beyond a pre-defined distance from a large proportion of data. However, finding a distance threshold for explicitly identifying outliers is challenging. Therefore, algorithms that compute an outlier score for each data point were developed that do not rely on a fixed distance for labeling outliers. These methods are often categorized into *local* and *global* in the literature [4]. A classic global method is *k-Nearest Neighbor* ( $k$ NN) [10], where the outlier score of a data point is the distance to its  $k^{th}$  nearest neighbor ( $k$ NN–dist). In  $k$ NN the potential outliers are expected to have large  $k$ NN–dist values and can be considered “global”

as their deviation is measured w.r.t. the overall dataset. On the other hand, the Local Outlier Factor (LOF) [11] is another classic method that finds outliers by assessing the density deviation of each data point relative to the deviation of its nearest neighbors (local neighborhood) used as the reference set. These outliers are considered “local” as they are outliers w.r.t. their nearest neighbors. Thus, choosing an appropriate method depends on the type of outliers present in the data, which is challenging to know beforehand. Differently, GLOSH proposes a way to dynamically select the reference sets based on the HDBSCAN\* hierarchy, offering a more adaptive approach to detect both types of outliers [5].

However, GLOSH requires the  $min_{pts}$  parameter indirectly since it is required by HDBSCAN\* to construct the cluster hierarchy, which is known to be sensitive; even small changes in the  $min_{pts}$  value can lead to very different data partitions [12]. Swersky [13] systematically showed that the performance of GLOSH can significantly vary for different  $min_{pts}$  values. While automatic methods exist for extracting flat clustering results from hierarchies [12], [14], there is no prior research for the automatic selection of the  $min_{pts}$  value for hierarchical clustering methods. Finding the best  $min_{pts}$  value for GLOSH in combination with HDBSCAN\* is challenging in unsupervised scenarios, as there is no prior knowledge about the underlying distribution of the data. Recent works such as [15] uses meta learning to select outlier detection models for a new dataset based on its past performance on similar datasets. Different from [15], we use a single outlier detection model, GLOSH, and do not select it from a set of models. Additionally, our method does not require prior information about the dataset or the past performance of GLOSH on similar datasets. Such prior information, while advantageous, may not always be available.

## III. BACKGROUND

### A. HDBSCAN\*

HDBSCAN\* or *Hierarchical DBSCAN\** [5] is a hierarchical way of clustering data which is an improvement of one of the most popular density-based clustering algorithms, DBSCAN [16]. In DBSCAN\*, density-based clusters are constructed using two parameters: (I) the radius  $\epsilon$ , and (II) the minimum number of points  $min_{pts}$ . A point  $x_i$  in a dataset  $D$  is a *core point* if it has  $min_{pts}$  data points in its  $\epsilon$ -neighborhood  $N_\epsilon(x_i)$  (i.e. in the set of points that are located less than or equal to distance  $\epsilon$  away from  $x_i$ ); otherwise, a point is labeled as noise. Two *core points* are *density connected* w.r.t.  $\epsilon$  and  $min_{pts}$  if they directly or transitively belong to each other’s  $N_\epsilon$ . DBSCAN\* defines a cluster as a maximal set of points where each pair of points in the set are *density connected*.

HDBSCAN\* improves DBSCAN\* and does not require the user to pre-define an  $\epsilon$  radius. For a  $min_{pts}$  value, HDBSCAN\* computes (I) the *core distance*  $\epsilon_c(x_i)$  for each  $x_i \in D$ , which is the minimum  $\epsilon$  distance required for  $x_i$  to be a *core point*, and (II) the *mutual reachability distance*  $d_{mrd}(x_i, x_j)$ , which is the minimum distance required for  $x_i$

and  $x_j$  to be in each other's  $\epsilon$ -neighborhood while both are also *core points*. The  $d_{mrd}(x_i, x_j)$  is defined as:

$$d_{mrd}(x_i, x_j) = \max\{\epsilon_c(x_i), \epsilon_c(x_j), d(x_i, x_j)\} \quad (1)$$

where  $d(x_i, x_j)$  represents the underlying distance between the two points in the dataset. The mutual reachability distance  $d_{mrd}$  is dependent on the value of  $min_{pts}$ . The HDBSCAN\* hierarchy w.r.t. a single  $min_{pts}$  value is obtained by computing the *minimum spanning tree*  $MST_{min_{pts}}$  of a complete, edge-weighted, virtual graph  $G_{min_{pts}}$  [8], where the vertices represent the data points in the dataset, and the edge weights are the values of the *mutual reachability distance*  $d_{mrd}$  between them. The edges from the MST are then removed in decreasing order of edge weights. At every removal, we obtain a new set of connected core objects (which are the clusters) and the remaining noise (points that are not connected). This creates a hierarchy of all possible DBSCAN\* solutions obtained at  $\epsilon \in [0, \infty]$  for a given  $min_{pts}$  value.

## B. GLOSH

The HDBSCAN\* framework includes the *Global-Local Outlier Scores based on Hierarchies* (GLOSH) algorithm to detect outliers using hierarchical density estimates. GLOSH computes a score for each data point in a dataset w.r.t. a reference set of points dynamically selected from the cluster structure within the HDBSCAN\* hierarchy.

The GLOSH score for a point  $x_i \in D$  is based on  $C_{x_i}$ , which is the cluster where  $x_i$  is assigned to for the minimum  $\epsilon$  value in the hierarchy that makes  $x_i$  a core point. The density of  $x_i$ ,  $\lambda(x_i)$  is compared to that of the densest point in  $C_{x_i}$ ,  $\lambda_{max}(C_{x_i})$ , which is the point that is assigned to the  $C_{x_i}$  when it is first formed in the hierarchy (making it the longest surviving point in  $C_{x_i}$ ). The GLOSH score  $\Gamma_{min_{pts}}(x_i)$  of  $x_i$  for a particular  $min_{pts}$  value is defined as:

$$\Gamma_{min_{pts}}(x_i) = \frac{\lambda_{max}(C_{x_i}) - \lambda(x_i)}{\lambda_{max}(C_{x_i})} \quad (2)$$

where  $\lambda$  is defined as  $\frac{1}{\epsilon_c(x_i)}$  for any point  $x_i$ . The  $\lambda_{max}(C_{x_i})$  is used as the referential density to compare the density of any point  $x_i$  that has  $C_{x_i}$  as its original cluster in the hierarchy. The GLOSH score falls in the range  $[0, 1]$ . If  $x_i$  resides in a dense region of a cluster, then  $\lambda_{max}(C_{x_i}) - \lambda(x_i)$  tends to be 0 whereas, for points that reside in low-density areas,  $\lambda_{max}(C_{x_i}) - \lambda(x_i)$  tends to increase, resulting in higher GLOSH scores. A pre-defined threshold is applied to the GLOSH scores to label outliers.

Thus, GLOSH depends on two parameters that are often unknown in practice: the  $min_{pts}$  for computing the point densities and the threshold for labeling outliers. In section IV, we propose a way to find the “best”  $min_{pts}$  value, and in section V, a strategy to determine the threshold.

## IV. USING GLOSH-PROFILES FOR $min_{pts}$ SELECTION

This section discusses selecting the best  $min_{pts}$  value for GLOSH in four parts: (1) introducing GLOSH-Profiles as a tool to find the best  $min_{pts}$  value (IV-A), (2) linking the

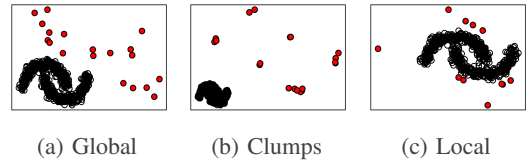


Fig. 1: Banana Dataset with different kinds of outliers.

uniform rate of change in profiles to the best  $min_{pts}$  value (IV-B and IV-C), (3) proposing Auto-GLOSH to automate the selection from the profiles (IV-D), and (4) outlining the complexity of Auto-GLOSH (IV-E).

### A. GLOSH-Profile

This sub-section introduces the notion of a GLOSH-Profile that captures the behavior of the GLOSH scores of a data point across different HDBSCAN\* hierarchies over a range of  $min_{pts}$  values. The impact of changing the  $min_{pts}$  values on cluster formation has been showed in prior studies [17], [8]. An increase in  $min_{pts}$  values requires each point to have more points in its  $\epsilon$ -neighborhood to become a *core point*, resulting in larger core distance  $\epsilon_c(\cdot)$  values. In turn, larger core distances tend to form “smoother” (and possibly larger) clusters in the hierarchy. Conversely, decreasing  $min_{pts}$  values allows points to become *core points* with smaller  $\epsilon_c(\cdot)$  values, potentially building more detailed and smaller clusters within the base of the hierarchy. Therefore, considering different  $min_{pts}$  values, GLOSH may compare a point's density with that of different clusters (i.e., neighborhoods of different densities), which influences the score for that point. Consequently, different  $min_{pts}$  values may yield different results, and it is challenging to know which is the best. To address the problem of finding a single best value for  $min_{pts}$  for GLOSH we formally define a GLOSH-Profile as follows:

**Definition 4.1 (GLOSH-Profile).** A *GLOSH-Profile*  $P\Gamma_{m_{max}}(x_i)$  is an array of GLOSH scores  $\Gamma_{min_{pts}}(x_i)$  for a given point  $x_i$  for a range  $[2, m_{max}]$  of  $min_{pts}$  values between 2 and a maximum value  $m_{max}$ :

$$P\Gamma_{m_{max}}(x_i) = \begin{bmatrix} \Gamma_2(x_i) \\ \Gamma_3(x_i) \\ \vdots \\ \Gamma_{m_{max}}(x_i) \end{bmatrix} \quad (3)$$

The  $m_{max}$  value is in practice typically chosen to be smaller than 100 [18]. In this paper, when  $m_{max}$  is clear from the context or irrelevant for the discussion, we abbreviate  $P\Gamma_{m_{max}}(\cdot)$  as  $P\Gamma(\cdot)$ . We use GLOSH-Profiles as a guide to select the “best”  $min_{pts}$  value.

### B. GLOSH Performance across $min_{pts}$ values

We analyze the performance of GLOSH across various  $min_{pts}$  values to find the specific ranges where GLOSH achieves its best performance. Additionally, we compare these results with the GLOSH-Profiles as we aim to find the  $min_{pts}$  values that lead to the best GLOSH performance.

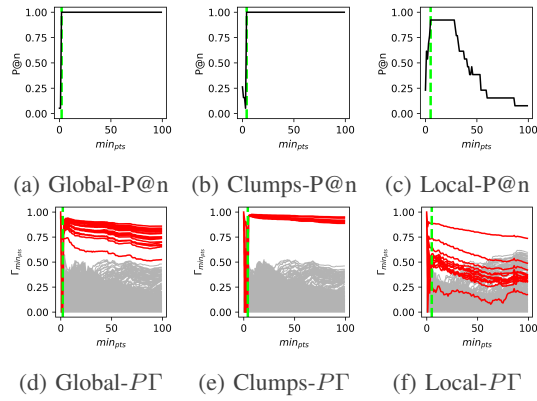


Fig. 2: Comparing GLOSH-Profiles ( $PT$ ) with Precision@ $n$  ( $P@n$ ) obtained by GLOSH at different  $min_{pts}$  values: The black line in Fig. 2a to 2c shows the  $P@n$  at every  $min_{pts}$  value. The outliers profiles are in red while the inliers profiles are in gray. The green (dashed) line denotes the specific  $min_{pts}$  value where GLOSH first achieves the best  $P@n$  within the range  $[2, 100]$  of  $min_{pts}$  values.

The performance of GLOSH is assessed using the Precision@ $n$  ( $P@n$ ) metric; it measures the fraction of “true outliers” among the points with the  $n$ -th highest GLOSH scores, where  $n$  is taken as the total number of labeled outliers in a test dataset. In this section, we use two-dimensional datasets to allow the visualization of the data distribution, facilitating an intuitive understanding of the results. We use the dataset with banana-shaped clusters from [19], commonly used for outlier detection studies in the literature [4]. To investigate outliers with different spatial characteristics, we followed [20] to generate different kinds of synthetic outliers such as local, global, and small groups of points (i.e., “outlier clumps” [21]) with similar behavior among themselves but dissimilar from the rest of the data. To generate local outliers, we learn a Gaussian Mixture Model (GMM) on the inlier samples. The covariance matrix  $\Sigma$  of the inlier samples is estimated using the GMM model and is scaled to  $\hat{\Sigma} = \alpha\Sigma$  (where  $\alpha = 5$  as in [20]). We use the GMM model to generate the local outlier samples using  $\hat{\Sigma}$ . To generate outlier clumps, we use the GMM to estimate the mean feature vector  $\mu$  of the inlier class samples. Then, we scale the mean feature vector by a factor of 5 ( $\hat{\mu} = \alpha\mu$  where  $\alpha = 5$  as in [20]) to generate the clumps using the GMM model. Hereafter, we will refer to these outliers as ‘Clumps’. We generate global outliers from a uniform distribution with boundaries defined by  $(\alpha \cdot \max(D^m), \alpha \cdot \min(D^m))$  where  $D^m$  represents the  $m$ -th feature of the inlier class data. Then, we apply the *tomek links* technique [22] to filter out generated points that fall “too close” to the existing inlier points, which would not characterize them as global outliers. An inlier and outlier data point forms a *tomek link* when they are the closest neighbors to each other. *Tomek link* indicates that the generated outlier falls inside or is located close to the border of a cluster and should be removed. Fig. 1 shows the datasets, including the generated outliers.

We make the following key observations. *First*, Fig. 2d to 2f show that for all the different kinds of outliers, the GLOSH scores in the outlier profiles fluctuate for low  $min_{pts}$  values, i.e., the GLOSH scores change significantly between consecutive  $min_{pts}$  values and at different rates for different profiles. This happens when several outliers get lower GLOSH scores than inliers. Therefore, in Fig. 2a to 2c, GLOSH results in small  $P@n$  values for low  $min_{pts}$ . *Second*, beyond a specific  $min_{pts}$  value, the GLOSH scores in the profiles start undergoing minimal changes between consecutive  $min_{pts}$  values. When they change, they do it at similar rates so that the resulting rankings tend to not change in a way that affects  $P@n$ . *Third*, for all kinds of outliers, we observe that the best  $P@n$  results correspond to the  $min_{pts}$  values where the GLOSH scores start changing at a similar rate. However, there is an exception for local outliers, as one cannot randomly choose a value beyond this specific  $min_{pts}$  value and expect to achieve the best  $P@n$  in the presented experiment. Fig. 2f shows that local outlier profiles are more similar to inlier profiles when compared to the profiles of global outliers and clumps. There is no notable gap between the profiles of local outliers and inliers, as local outliers are close to the inliers. Therefore, as the  $min_{pts}$  value increases beyond the value where the GLOSH scores start changing at a similar rate, some local outliers end up getting scores lower than several inliers. Overall, the experiments suggest that the  $min_{pts}$  value where GLOSH scores start changing at a similar rate in the profiles can potentially yield the best results. Based on these insights, we develop a way to find this  $min_{pts}$  value in the following.

### C. Dissimilarity between GLOSH score rankings for consecutive $min_{pts}$ values

Based on our findings in sub-section IV-B, in this sub-section we study how GLOSH scores change between consecutive  $min_{pts}$  values. One way to do that is by measuring the differences in the outlier rankings at consecutive  $min_{pts}$  values. However, ordering of the data points w.r.t. GLOSH scores may be problematic. Two points in the same density region within a cluster may get identical GLOSH scores at consecutive  $min_{pts}$  values. If one ranks the data points based on their GLOSH scores, any specific ordering of those two data points may differ between the consecutive  $min_{pts}$  values. In such cases, the ordering of many inliers will dominate any measure of data point rank dissimilarity. For outlier detection, we are primarily interested in outstanding score values or a small portion of the highest scores, as they often represent outliers. If points get identical GLOSH scores at consecutive  $min_{pts}$  values, they should not reflect the difference between the rankings. However, even though the ordering of points may differ, the scores of adjacent points in large stretches of a ranking may be very similar. Therefore, instead of measuring the dissimilarity between sequences of ordered points, we measure the dissimilarity between the sorted sequences of GLOSH scores obtained at consecutive  $min_{pts}$  values.

To measure the dissimilarity between the sorted GLOSH score sequences we use Pearson correlation, a popular sym-

metric metric that can measure the relationship between two outlier score sequences [23]. We can express dissimilarity by subtracting the correlation from 1. For two sorted GLOSH score sequences obtained at  $min_{pts} = k$  ( $S_k$ ) and  $min_{pts} = l$  ( $S_l$ ), the Pearson dissimilarity  $\Delta$  is defined as:

$$\Delta(S_k, S_l) = 1 - \left| \frac{Cov(S_k, S_l)}{\sqrt{Var(S_k)Var(S_l)}} \right| \quad (4)$$

where  $Cov$  measures the covariance between sorted GLOSH score sequences for  $min_{pts} = k$  and  $min_{pts} = l$ , and  $Var$  measures the variance of the values in each of the sequences. When most GLOSH-Profiles start changing at a similar rate, the resulting relative order and magnitude of GLOSH scores tend to undergo minimal changes between consecutive values of  $min_{pts}$ . Therefore, the Pearson dissimilarity is unaffected by absolute value changes, as it focuses on the means' variations. However, for lower  $min_{pts}$  values where the GLOSH scores in the outlier profiles may fluctuate, the covariance between the sorted GLOSH score sequences at consecutive  $min_{pts}$  values should be high. Thus, we expect a high Pearson dissimilarity between the sorted GLOSH score sequences for low consecutive  $min_{pts}$  values. Differently, for higher consecutive  $min_{pts}$  values, the scores vary at a similar rate, and the Pearson dissimilarity tends to be 0.

To study the behavior of GLOSH score rankings in more detail, we formally define the GLOSH Outlier Rank Dissimilarity-Profile (ORD-Profile)  $R_{m_{max}}$ . The ORD-Profile quantifies the dissimilarity between the sorted sequences of GLOSH scores at consecutive  $min_{pts}$  values as they increase. **Definition 4.2. GLOSH Outlier Rank Dissimilarity-Profile:** The GLOSH ORD-Profile  $R_{m_{max}}$  is an array of dissimilarity values for pairs of sorted sequences of GLOSH scores  $S_{min_{pts}}$  obtained at consecutive  $min_{pts}$  values in a range  $[2, m_{max}]$ :

$$R_{m_{max}} = \begin{bmatrix} \Delta(S_2, S_3) \\ \Delta(S_3, S_4) \\ \vdots \\ \Delta(S_{m_{max}-1}, S_{m_{max}}) \end{bmatrix} \quad (5)$$

As we define GLOSH-Profiles from  $min_{pts} = 2$ , each dissimilarity score  $r_i \in R_{m_{max}}$  is obtained between  $min_{pts} = i + 2$  and  $min_{pts} = i + 3$ , where the indices  $i$  starts from 0.

Fig. 3 compares ORD-Profiles with Precision@n (P@n) for a range of  $min_{pts}$  values. *First*, Fig. 3d to 3f show a high dissimilarity in the ORD-Profile  $R_{m_{max}}$  for lower  $min_{pts}$  values, that evolve to 0 and form an “elbow like” structure. The “elbow” is the point from where GLOSH scores start changing at a similar rate in most profiles. *Second*, the  $min_{pts}$  value where the “elbow” is formed corresponds to the  $min_{pts}$  value where GLOSH achieves its best P@n. As local outliers are close to inliers, they do not get GLOSH scores significantly higher than inliers. This happens as some local outliers end up getting lower GLOSH scores as inliers, as discussed in section IV-B. For global outliers and clumps, GLOSH scores are higher than inliers for a large range of  $min_{pts}$  values. In summary, Fig. 3 suggests that choosing the  $min_{pts}$  value directly at the “elbow” may potentially yield the best P@n.

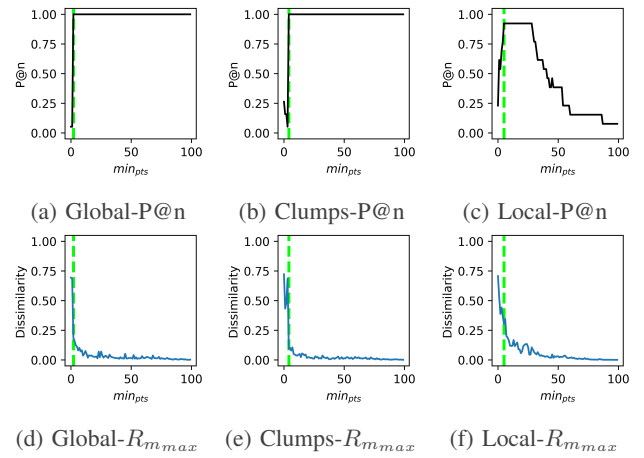


Fig. 3: Comparing the ORD-Profiles ( $R_{m_{max}}$ ) with GLOSH's P@n across different  $min_{pts}$  values. We highlighted (with a dashed line) the specific  $min_{pts}$  value where GLOSH first records the best P@n within a range  $[2, 100]$  of  $min_{pts}$  values.

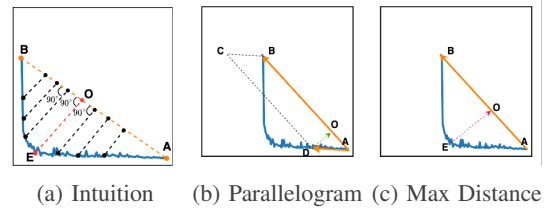


Fig. 4: Illustrating the process of finding the Elbow of the Outlier Rank Dissimilarity-Profile on the Banana Dataset with Global Outliers

#### D. Finding the “Best” $min_{pts}$ value

Here, we propose Auto-GLOSH to find the  $m^*$  value, i.e., the  $min_{pts}$  value at the “elbow” in the ORD-Profile and use it to yield the best results for GLOSH, as in sub-section IV-C. **Intuition.** Fig. 4a illustrates the intuition behind Auto-GLOSH. Assuming a line AB is drawn connecting the maximum value (B) and end value (A) of an ORD-Profile  $R_{m_{max}}$ , the latter with index  $|R_{m_{max}}|$ . Perpendicular lines from AB to each value in the  $R_{m_{max}}$  represent the “shortest distance” (orthogonal distance) between those values and AB. In this context, the value at the “elbow” (E) is the one that has the largest orthogonal distance (shown as EO).

**Method.** Given an ORD-Profile,  $R_{m_{max}}$ , find the maximum dissimilarity value in  $R_{m_{max}}$  and call it  $b$ . Consider  $i$  the index of  $b$  in  $R_{m_{max}}$ , which corresponds to the largest peak in the ORD-Profile plot in Fig. 4. As shown in Fig. 4b and Fig. 4c, we compute a vector  $\vec{AB}$  starting from the end value in  $R_{m_{max}}$ ,  $A = (|R_{m_{max}}|, R_{m_{max}}[|R_{m_{max}}|])$ , and the maximum value,  $B = (i, b)$ , of the ORD-Profile. Then, we compute  $\vec{AB}$  as  $B - A$ . We will use  $\vec{AB}$  as a tool to find the value in the ORD-Profile that has the largest orthogonal distance to  $\vec{AB}$ .

Next, we compute the orthogonal distances between each dissimilarity value lying between A and B in the ORD-Profile, and  $\vec{AB}$ . To do that, consider the parallelogram ABCD as in

Fig. 4b where  $D$  is a value between  $A$  and  $B$  in ORD-Profile. Assuming  $\overrightarrow{AB}$  as the base of the parallelogram  $ABCD$ , the orthogonal distance between  $D$  and  $\overrightarrow{AB}$ , represented as  $\overrightarrow{DO}$ , is the height of  $ABCD$ . Our goal is to find the  $\|\overrightarrow{DO}\|$ , where  $\|\cdot\|$  is the magnitude. In classical geometry, the area of a parallelogram is computed as *base*  $\times$  *height*, which is  $\|\overrightarrow{AB}\| \times \|\overrightarrow{DO}\|$  w.r.t. Fig. 4b. As shown in [24], [25], using vectors, the area of the parallelogram  $ABCD$  is:

$$\|\overrightarrow{AD} \times \overrightarrow{AB}\| = \|\overrightarrow{AB}\| \times \|\overrightarrow{DO}\| \implies \|\overrightarrow{DO}\| = \frac{\|\overrightarrow{AD} \times \overrightarrow{AB}\|}{\|\overrightarrow{AB}\|} \quad (6)$$

where,  $\overrightarrow{AD}$  is computed as  $D - A$ . We compute the orthogonal distances of each dissimilarity value lying between  $A$  and  $B$  in the ORD-Profile and find the value with the largest distance as the elbow point. This can be done efficiently by continuously tracking the largest distance as we compute the orthogonal distances of each dissimilarity value. Assuming  $r_i$  as the value at the elbow (discussed in sub-section IV-C), we can find  $r_i$  at  $\min_{pts} = i + 3$ , on measuring the dissimilarity between the sequences at  $\min_{pts} = i + 3$  and  $\min_{pts} = i + 2$ . We take  $i + 3$  as the value of  $m^*$ .

### E. Complexity Analysis

Our method extracts the GLOSH-Profiles from different HDBSCAN\* hierarchies at different  $\min_{pts}$  values. The naïve way to build these profiles is to run HDBSCAN\* for every  $\min_{pts}$  value up to  $m_{max}$ . This may be inefficient and require  $\mathcal{O}(n^2 m_{max})$  time as a single HDBSCAN\* run takes  $\mathcal{O}(n^2)$ , assuming  $n$  is the dataset size and  $m_{max} \ll n$  ( $m_{max}$  is typically chosen to be below 100 in practice). However, *CORE-SG* permits a more efficient process [8]. With a single HDBSCAN\* run at  $\min_{pts} = m_{max}$ , the *CORE-SG* $_{m_{max}}$  can be build from  $MST_{m_{max}}$  and  $m_{max}$ -Nearest Neighbor Graph [8]. By replacing the complete graph with the *CORE-SG* $_{m_{max}}$ , all  $MST_{\min_{pts}}$  with  $\min_{pts}$  values up to  $m_{max}$  can be obtained with an asymptotic computational complexity of  $\mathcal{O}(nm_{max}^2 \log n)$ , which is lower than a single run of HDBSCAN\* for  $m_{max} \ll n$ .

As shown in [5], the GLOSH scores for all the datapoints can be computed in  $\mathcal{O}(n)$  time using the  $MST_{\min_{pts}}$  and equation 2. From the scores, the GLOSH-Profiles can be built in  $\mathcal{O}(nm_{max})$  time. From the profiles, the Pearson dissimilarity is measured between consecutive GLOSH score sequences. Expressed as  $\Delta(\cdot, \cdot)$  in equation 4, calculating the dissimilarity requires  $\mathcal{O}(n)$  time, as measuring *Cov* and *Var* requires iterating over each GLOSH score in the sequences. Therefore, building the ORD-Profile  $R_{m_{max}}$  needs  $\mathcal{O}(nm_{max})$  time. Then, assessing the  $m^*$  value has  $\mathcal{O}(n)$  time, as finding the maximum value in  $R_{m_{max}}$  and obtaining the largest orthogonal distance are procedures with linear cost in relation to  $n$ .

In summary, the overall asymptotic computational cost of our method is upper bounded by the first run of HDBSCAN\* to build the *CORE-SG* $_{m_{max}}$ , which is  $\mathcal{O}(n^2)$ . Note that efficient and parallel algorithms for *MST* construction and cluster

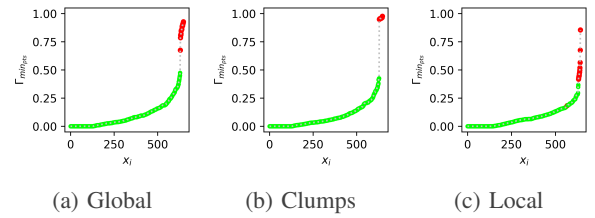


Fig. 5: Banana Dataset: Sorted Sequence of GLOSH scores at  $\min_{pts} = m^*$ . The x-axis presents the data points  $x_i$  in ascending order of their GLOSH scores, and the y-axis represents the corresponding GLOSH scores  $\Gamma_{m^*}(x_i)$ . The green dots represent the GLOSH scores of inliers, and the red represents that of outliers.

extraction in HDBSCAN\* exist, which can further accelerate our method. For instance, [26] shows that HDBSCAN\*'s complexity can be reduced to  $\mathcal{O}(n \log^2 n)$  in sequential time and  $\mathcal{O}(\log^2 n \log \log n)$  in parallel time. Another approach based on data summarization and parallelism performed well for large datasets [27].

## V. UNSUPERVISED LABELLING OF POTENTIAL OUTLIERS

Here, we address the problem of choosing a threshold for labelling inliers and potential outliers. We first study the distribution of the GLOSH scores at the  $m^*$  (V-A), then we design POLAR to find the threshold (V-B).

### A. GLOSH scores at the $m^*$

In practice, the number of outliers is usually unknown, which may be challenging to estimate. We believe that a pattern or gap in the sorted sequence of GLOSH scores at the  $m^*$  value can indicate a suitable threshold for this value. A “gap” would be a noticeable deviation or abrupt increase in the sorted sequence, suggesting a potential decrease in the density of points and indicating the possibility of outliers.

The example in Fig. 5 illustrates the intuition behind the threshold estimation we propose. As expected, outliers receive high GLOSH scores and are predominantly located at the end of the sequence for the assessed  $m^*$  value. One can see that the sorted GLOSH score sequences form a “knee” when the scores start deviating from the initial linear trend and show accelerated growth. This is true for all kinds of outliers in Fig. 5a to 5c. Such a “knee” is formed by GLOSH scores of points that significantly deviate from the majority of points and may serve as a suitable threshold to label potential outliers. However, as shown in Fig. 5, there may be a few points with high deviations and GLOSH scores beyond the “knee” that behave like outliers but are labeled as inliers. To avoid classifying these points as outliers, it is necessary to adjust the selection of the threshold “beyond the knee” of the sorted sequence. Based on these insights, we present our threshold estimation approach in the following sub-section.

### B. Automatically Labelling Potential Outliers

In this sub-section, we design POLAR, a fully unsupervised approach to label potential outliers by using the sorted GLOSH

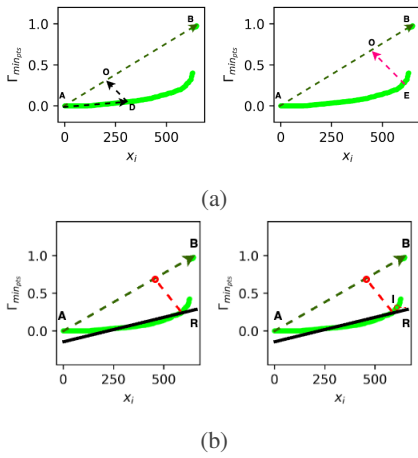


Fig. 6: Banana Dataset with clumps: **(a)** Finding the “knee” point in the sorted sequence of GLOSH scores at  $min_{pts} = m^*$ . The “knee” GLOSH score  $E$  has the maximum orthogonal distance to  $\overrightarrow{AB}$ . **(b)** Adjusted threshold estimation at  $m^*$ .

score sequences obtained at  $m^*$  value. In the first step, our approach finds the “knee” in the GLOSH score sequences at the  $m^*$  value. However, to avoid incorrectly labeling the inliers that may lie beyond the “knee”, an adjustment strategy is necessary for choosing the threshold.

**Finding the “knee” GLOSH score:** To find the “knee”, we use a similar strategy to the one previously presented in section IV-D to locate the “elbow” in the ORD–Profile. As illustrated in Fig. 6a, we firstly compute the vector  $\overrightarrow{AB}$ , as the difference between the last GLOSH score  $B$  and the first GLOSH score  $A$  in the sorted sequence. Next, we compute the orthogonal distances between each GLOSH score and  $\overrightarrow{AB}$ . The GLOSH score in the sequence with the maximum orthogonal distance is identified as the “knee” GLOSH score.

**Adjusting the Inlier Threshold:** To adjust a threshold beyond the “knee” GLOSH score in the sorted sequence, we first aim to capture the progression of the inlier GLOSH scores in the sorted sequence. We use a simple linear regression model [28] to estimate the trend of the inlier GLOSH scores in the sequence. The idea here is to capture the “almost” linear trend of the GLOSH scores observed at the beginning of the sorted sequence for inliers. The linear regression model estimates a quantitative value  $\hat{Y}$  based on a predictor variable  $X$ , assuming there is approximately a linear relationship between  $X$  and  $Y$ . The linear regression model can be represented as:

$$\hat{Y} = \beta_0 + \beta_1 \times X \quad (7)$$

For a dataset with  $N$  points, the indexes of each GLOSH score in the sorted sequence can be represented as  $[x_1, x_2, \dots, x_N]$ . We use the GLOSH scores before the “knee” GLOSH score in the sorted sequence to estimate the model coefficients  $\beta_0$  and  $\beta_1$ . The optimal values for  $\beta_0$  and  $\beta_1$  are estimated by minimizing the Mean Squared Error (MSE) between the estimated  $\hat{Y}$  and the known  $Y$ . In Fig. 6b, the estimated regression line is represented as a black solid line. Using the

regression model, we estimate the highest GLOSH score  $R$  at index  $x_N$ . Intuitively, this estimated GLOSH score  $R$  reflects the score that could have been reached if there were only inliers in the dataset and the scores in the sequence followed the trend that was observed up to the “knee”. Then, we search the GLOSH score,  $I$ , that is the most similar to  $R$  in the sorted sequence between the “knee” and  $x_N$  and choose it as the adjusted threshold.

## VI. EXPERIMENTAL ANALYSIS

### A. Dataset

To evaluate Auto-GLOSH and POLAR, we use a total of 69 datasets. This includes the Banana dataset from sub-section IV-B, along with two additional datasets, Anisotropic and Circular, obtained from [19]. Different types of outliers (global, local, clumps) are generated on these datasets, following the approach described in sub-section IV-B. Additionally, we make semi-synthetic datasets by using the real inliers from real one class classification (OCC) datasets and generating synthetic outliers on them. To do this, we use fifteen different real OCC datasets [20] that are popularly used in the literature: MVTec-AD\_zipper, HEPATITIS, LETTER, PIMA, STAMPS, VERTEBRAL, VOWELS, WDBC, WINE, WPBC, YEAST, BREASTW, CARDIO, CARDIOTOCOGRAPHY, and 20news\_3. They are across various domains: image, healthcare, documents, biology, chemistry, and NLP.

**Why synthetic outliers?** Firstly, nonsynthetic datasets commonly do not have their “real outliers” (as statistically defined in the literature [1]) previously labeled and available for analysis. For this reason, it is usual to rely on real OCC datasets, making them popular in outlier detection studies. These are originally classification datasets where one class is downsampled to fit into an outlier detection setting, assuming that the downsampled class (re-labeled as “outlier class”) has the traditional characteristics of “real outliers”. The characteristics of these “labeled outliers” (data points in the “outlier class”) are not fully understood in the literature [29]. This provides limited scope for testing our methods on different types of outliers as the specific outliers types that are present in the so-called “outlier class” are often unknown. We aim to evaluate our methods across different outlier types, which is not possible using the real OCC datasets.

Secondly, as also discussed in [29], in many real OCC datasets the “labeled outliers” may often deviate from the characteristics typically associated with outliers in the outlier detection literature (global, local, clumps). To support this argument, we report in Table I the best Precision@n (P@n) obtained on real OCC datasets of state-of-the-art neighborhood-based methods, GLOSH, KNN, and LOF, and parameter-free methods, ABOD [30] and COPOD [31]. The overall results in Table I reflect that across most of the datasets, none of the methods work well in identifying the “labeled outliers”, even with their best parameters. This indicates that many of the “labeled outliers” do not have the properties typically associated with the way these state-of-the-art methods define outliers. Additionally, we illustrate the behavior of the “labeled

Dataset	GLOSH	KNN	LOF	ABOD	COPOD
MVTec-AD_zipper	.62	.62	.6	.6	.58
HEPATITIS	.3	.3	.3	.23	.46
LETTER	.3	.39	.54	.1	.04
PIMA	.55	.55	.51	.55	.48
STAMPS	.25	.25	.19	.16	.41
VERTEBRAL	.03	.03	.1	.03	0
VOWELS	.54	.54	.38	.34	0
WDBC	.5	.5	.6	.4	.8
WINE	.4	.4	.4	.1	.4
WPBC	.19	.19	.21	.19	.21
YEAST	.28	.27	.32	.26	.25
BREASTW	.94	.94	.32	.95	.94
CARDIO	.53	.53	.28	.44	.52
CARDIOTOCOGRAPHY	.41	.4	.35	.38	.36
20news_3	.13	.13	.2	.13	.13

TABLE I: Results on real one-class classification datasets. For GLOSH, KNN, and LOF, we report the best results obtained across the commonly used  $min_{pts}$  and  $k$  values in the literature and practice [5], [32], [33], [34], [35].

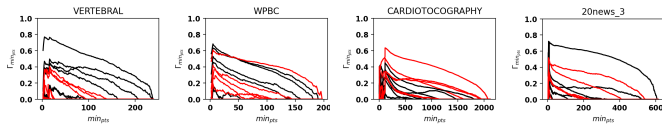


Fig. 7: GLOSH-Profiles of real one-class classification datasets [20]: VERTEBRAL, WPBC, CARDIOTOCOGRAPHY, and 20news\_3. To have a clear view of the individual profiles, we randomly choose six inlier and outlier class data points and plot their profiles. The red profiles represent the outlier class and the black represents the inlier class.

outliers” through GLOSH-Profiles to allow us to understand the properties of these data points better. Fig. 7 shows that the profiles of many “labeled outliers” achieve GLOSH scores lower than the “labeled inliers”, as such points sometimes behave like inlier profiles. To derive meaningful conclusions in our investigations, it is essential to have outliers that conform to their statistical definition [1], i.e., they exhibit characterizable deviations from the inliers.

Due to the aforementioned reasons, we generated synthetic outliers following the approach described in sub-section IV-B to evaluate Auto-GLOSH and POLAR. This approach to generating outliers is more meaningful than downsampling a class, as a generative model is used to create outliers with specific characteristics close to those of “real outliers”, as statistically defined in the literature.

### B. Auto-GLOSH

**Setup.** For evaluating Auto-GLOSH, we focus on three key questions: (I) Does the Auto-GLOSH estimated  $m^*$  value match the best GLOSH performance? (II) Can the estimated  $m^*$  value surpass or match results obtained with commonly used  $min_{pts}$  values in GLOSH? (III) Can GLOSH with the  $m^*$  value outperform existing state-of-the-art outlier detection methods? A comparative analysis is conducted between Auto-GLOSH and GLOSH with some commonly used  $min_{pts}$  values in the literature and practice. Performance is assessed using Precision@n (P@n). We also compare the performance at the  $m^*$  value with that achieved at the best  $min_{pts}$  value within range [2, 100], i.e., the value where GLOSH obtains its

highest P@n. Furthermore, we evaluate the performance at the  $m^*$  value against the state-of-the-art outlier detection methods such as LOF and KNN, which operate w.r.t. a neighborhood parameter  $k$ . Specifically, we compare the results obtained using Auto-GLOSH to those of LOF and KNN across commonly used  $k$  values. We also compare our results against parameter-free methods such as ABOD and COPOD and present a comparative analysis of the runtime required by our method.

**Auto-GLOSH Vs. GLOSH.** In Table II, for most datasets, our Auto-GLOSH estimated  $m^*$  value yields the best P@n that is achievable using GLOSH (represented as \*Best). Comparatively, using the commonly used  $min_{pts}$  values ( $\Gamma_5$  to  $\Gamma_{100}$ ) may result in a low P@n if the  $min_{pts}$  value is not chosen properly. Notably, for the Anisotropic dataset with local outliers, the Auto-GLOSH gets a higher P@n than all the commonly used  $min_{pts}$  values. For some datasets with local outliers, such as CARDIO or CARDIOTOCOGRAPHY, Auto-GLOSH did not match the best P@n; however, it is close to the best. This shows that Auto-GLOSH is able to find a  $min_{pts}$  value that can yield the best or nearly the best GLOSH results.

**Auto-GLOSH Vs. Neighborhood-based Methods.** In Table III, both KNN and LOF can outperform the best results achievable using GLOSH for some of the  $k$  values, especially for local outliers and clumps. Auto-GLOSH is built upon GLOSH and designed to extract the best results achievable with GLOSH. Consequently, it is expected that Auto-GLOSH may not surpass the performance of KNN and LOF when GLOSH itself does not outperform them. However, the overall results show that KNN and LOF only perform well for the best  $k$  values, i.e.,  $k$  must be chosen carefully. In LETTER with local outliers, increasing  $k$  decreases P@n for KNN but keeps it stable for LOF. For PIMA with local outliers, P@n decreases for both KNN and LOF as  $k$  increases. Examples like these make it unclear which  $k$  value to choose when the underlying data distribution is unknown. This emphasizes the necessity for Auto-GLOSH, which finds the  $min_{pts}$  value capable of achieving high P@n on most datasets.

**Auto-GLOSH Vs. Parameter-free Methods.** In Table IV, Auto-GLOSH consistently achieves high P@n across most datasets compared to ABOD and COPOD. Similar to Table III, there are instances, such as BREASTW with local outliers, where COPOD outperforms the best results of GLOSH. Consequently, it also outperforms Auto-GLOSH. As COPOD measures outlieriness along each dimension, it is able to identify the outliers that deviate only along certain dimensions. However, we argue that our method is the most reliable compared to the state-of-the-art parameter-free methods, as it records the highest P@n for most datasets across all kinds of outliers.

**Runtime Analysis.** Concerning the runtime of the algorithms presented in Table V, it is clear that by using *CORE-SG* we can achieve results much faster than running HDBSCAN\* at every  $min_{pts}$  value up to  $m_{max}$  (the naïve approach). This faster running time is achieved as the base *CORE-SG* graph is built just once, and the *MST* for each  $min_{pts}$  value is independently extracted from it. Although COPOD displayed the best runtime, Auto-GLOSH obtained a similar





Dataset	POLAR												*Best															
	Knee				G-Mean				Recall				F-measure				F-measure				G-Mean							
	Recall	F-measure			G-Mean			Recall			F-measure			G-Mean			F-measure			G-Mean								
G	C	L	M	G	C	L	M	G	C	L	M	G	C	L	M	G	C	L	M	G	C	L	M					
MVFac-AD_zipper	1	1	1	1	.60	.49	.27	.66	.96	.94	.97	.95	1	1	1	1	1	.87	.66	1	1	.99	.99	1	1	1	1	1
HEPATITIS	1	1	1	1	.66	.75	.11	.85	.97	.98	.57	.98	1	1	0	1	1	1	0	.92	1	1	0	.99	1	1	1	1
LETTER	1	1	1	1	.61	.61	.65	.78	.96	.96	.97	.97	1	1	.85	1	.98	.94	.91	1	.99	.99	.92	1	1	1	.92	1
PIMA	1	1	1	1	.7	.61	.29	.65	.97	.96	.88	.94	1	1	.9	1	.94	.96	.88	.67	.99	.99	.95	.94	1	1	.93	1
STAMPS	1	1	1	1	.34	.41	.40	.61	.9	.92	.93	.92	1	1	.35	1	.73	.79	.41	.66	.98	.98	.59	.94	1	1	.53	1
VERTEBRAL	1	1	1	1	.34	.37	.35	.55	.9	.91	.91	.91	1	1	.8	1	.68	.71	.61	.85	.97	.98	.87	.98	.95	1	.77	.97
VOWELS	1	1	1	1	.73	.46	.35	.66	.98	.94	.96	.95	1	1	.96	1	.95	.87	.74	1	.99	.99	.97	1	1	1	.92	1
WDBC	1	1	1	1	.36	.33	.08	.46	.9	.89	.91	.90	1	1	1	1	.85	.83	.3	.68	.99	.99	.98	.96	1	1	.85	.98
WINE	1	1	1	1	.44	.41	.26	.51	.93	.92	.95	.91	1	1	1	1	1	.92	.5	1	1	.99	.98	1	1	1	1	1
WPBC	1	1	1	1	.18	.34	.25	.45	.9	.89	.89	.89	1	1	1	1	.54	.89	.66	.83	.98	.99	.98	.98	1	1	.88	1
YEAST	1	1	.97	1	.32	.31	.33	.43	.89	.88	.88	.85	.75	.4	.97	.83	.69	.4	.77	.77	.86	.62	.97	.89	.73	.56	.91	.77
BREASTW	1	1	1	1	.23	.21	.16	.93	.82	.8	.71	.99	1	1	.72	1	.51	.46	.31	.65	.95	.94	.78	.94	1	1	.36	1
CARDIO	1	1	1	1	.42	.37	.39	.56	.93	.91	.92	.91	1	1	.71	.98	.91	.94	.73	.92	.99	.83	.97	.95	1	.85	.88	.94
CARDIOCTOGRAPHY	1	1	1	1	.65	.55	.48	.56	.97	.95	.94	.91	1	1	.79	1	.96	.96	.84	.98	.99	.99	.98	.99	1	1	.91	1
20news_3	1	1	1	1	.45	.43	.43	.61	.93	.93	.94	.93	1	1	1	1	.92	.75	.91	.91	.98	.98	.98	.99	1	1	.97	.99
Anisotropic	1	1	1	-	.3	.21	.24	-	.93	.88	.93	-	1	1	.92	-	.95	.86	.6	-	.99	.99	.94	-	1	1	.74	-
Banana	1	1	.92	-	.33	.45	.55	-	.93	.96	.94	-	1	1	.53	-	.92	1	.7	-	.99	1	.73	-	1	1	.96	-
Circular	1	1	.76	-	.77	.76	.68	-	.99	.99	.87	-	.89	1	.3	-	.94	1	.47	-	.94	1	.55	-	1	1	.86	-

TABLE VI: Evaluating POLAR on datasets with global outliers (G), clumps (C), local outliers (L), and mixed outliers (M). We compare with the best performance that can be achieved across all possible thresholds.

[3] C. Bellinger, S. Sharma, and N. Japkowicz, "One-class versus binary classification: Which and when?" in *2012 11th International conference on machine learning and applications*, vol. 2. IEEE, 2012, pp. 102–106.

[4] H. O. Marques, L. Swersky, J. Sander, R. J. Campello, and A. Zimek, "On the evaluation of outlier detection and one-class classification: a comparative study of algorithms, model selection, and ensembles," *Data Mining and Knowledge Discovery*, pp. 1–45, 2023.

[5] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 1, pp. 1–51, 2015.

[6] A. Cavalcante Araujo Neto, *A Framework for Hierarchical Density-Based Clustering Exploration*. Ph.D. thesis, University of Alberta, 2021.

[7] Z. Zuo, Z. Li, P. Cheng, and J. Zhao, "A novel subspace outlier detection method by entropy-based clustering algorithm," *Scientific Reports*, vol. 13, p. 15331, 2023.

[8] A. C. A. Neto, M. C. Naldi, R. J. Campello, and J. Sander, "Core-sg: Efficient computation of multiple msts for density-based methods," *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 951–964, 2022.

[9] E. M. Knorr and R. T. Ng, "A unified notion of outliers: Properties and computation." *KDD*, vol. 97, pp. 219–222, 1997.

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

[11] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

[12] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.

[13] L. Swersky, *A study of unsupervised outlier detection for one-class classification*. M.Sc. thesis, University of Alberta, 2018.

[14] R. J. Campello, D. Moulavi, A. Zimek, and J. Sander, "A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies," *Data Mining and Knowledge Discovery*, vol. 27, pp. 344–371, 2013.

[15] Y. Zhao, R. Rossi, and L. Akoglu, "Automatic unsupervised outlier model selection," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4489–4502, 2021.

[16] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[17] A. C. A. Neto, J. Sander, R. J. Campello, and M. A. Nascimento, "Efficient computation and visualization of multiple density-based clustering hierarchies," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 8, pp. 3075–3089, 2019.

[18] H. O. Marques, R. J. Campello, A. Zimek, and J. Sander, "On the internal evaluation of unsupervised outlier detection," in *Proceedings of the 27th international conference on scientific and statistical database management*, 2015, pp. 1–12.

[19] P. Koncar, "Synthetic dataset for outlier detection," Feb. 2018. [Online]. Available: <https://doi.org/10.5281/zenodo.1171077>

[20] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "Adbench: Anomaly detection benchmark," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 142–32 159, 2022.

[21] H. O. Marques, R. J. Campello, J. Sander, and A. Zimek, "Internal evaluation of unsupervised outlier detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 14, no. 4, pp. 1–42, 2020.

[22] B. Krawczyk, I. Triguero, S. García, M. Woźniak, and F. Herrera, "Instance reduction for one-class classification," *Knowledge and Information Systems*, vol. 59, pp. 601–628, 2019.

[23] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel, "On evaluation of outlier rankings and outlier scores," in *Proceedings of the 2012 SIAM international conference on data mining*. SIAM, 2012, pp. 1047–1058.

[24] H. Anton and C. Rorres, *Elementary linear algebra: applications version*. John Wiley & Sons, 2013.

[25] D. C. Lay, *Linear algebra and its applications*. Pearson Education India, 2003.

[26] Y. Gu, Z. Napier, Y. Sun, and L. Wang, "Parallel cover trees and their applications," in *Proceedings of the 34th ACM Symposium on Parallelism in Algorithms and Architectures*, 2022, pp. 259–272.

[27] J. A. d. Santos, T. I. Syed, M. C. Naldi, R. J. G. B. Campello, and J. Sander, "Hierarchical density-based clustering using mapreduce," *IEEE Transactions on Big Data*, vol. 7, no. 1, pp. 102–114, 2021.

[28] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, "Linear regression," in *An Introduction to Statistical Learning: With Applications in Python*. Springer, 2023, pp. 69–134.

[29] G. Steinbuss and K. Böhm, "Benchmarking unsupervised outlier detection with realistic synthetic data," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 4, pp. 1–20, 2021.

[30] 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*, 2008, pp. 444–452.

[31] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "Copod: copula-based outlier detection," in *2020 IEEE international conference on data mining (ICDM)*. IEEE, 2020, pp. 1118–1123.

[32] T. Jayasinghe, D. Dixon, M. S. Povich, B. Binder, J. Velasco, D. M. Lepore, D. Xu, S. Offner, H. A. Kobulnicky, L. D. Anderson *et al.*, "The milky way project second data release: bubbles and bow shocks," *Monthly Notices of the Royal Astronomical Society*, vol. 488, no. 1, pp. 1141–1165, 2019.

[33] G. Jarry, D. Delahaye, F. Nicol, and E. Feron, "Aircraft atypical approach detection using functional principal component analysis," *Journal of Air Transport Management*, vol. 84, p. 101787, 2020.

[34] H. O. Marques, A. Zimek, R. J. Campello, and J. Sander, "Similarity-based unsupervised evaluation of outlier detection," in *International Conference on Similarity Search and Applications*. Springer, 2022, pp. 234–248.

[35] P. Yin and M. Peng, "Station layout optimization and route selection of urban rail transit planning: A case study of shanghai pudong international airport," *Mathematics*, vol. 11, no. 6, p. 1539, 2023.

[36] Y. S. Aurelio, G. M. De Almeida, C. L. de Castro, and A. P. Braga, "Learning from imbalanced data sets with weighted cross-entropy function," *Neural processing letters*, vol. 50, pp. 1937–1949, 2019.