A Novel Approach for Unveiling Layered Geometric Patterns in Noisy Unsupervised Data: A Study on Drug-Like Molecules

Luiz C. D. Cavalcanti[1], **Ricardo A. Rios**[1], **Tiago J. S. Lopes**[2], **Tatiane N. Rios**[1]

[1]Institute of Computing – Federal University of Bahia
Salvador, 40170-110, Brazil

[2]Nezu Biotech GmbH
Tiergartenstr. 15, 69121 Heidelberg, Germany

luizcdc@ufba.br, ricardoar@ufba.br, tatiane.nogueira@ufba.br

tiago.lopes@nezubiotech.com

Abstract. *Identifying meaningful examples in datasets where most or all data belong to a single category is a common challenge in Machine Learning (ML). In many real-world scenarios, such as in science, medicine, and industry, data for the target class is often abundant, while data from other classes is scarce or missing. This makes it difficult for ML models to differentiate between what belongs to the target class and what does not. This paper presents a novel approach to address this issue by combining clustering and geometric analysis techniques. We develop a one-class classification method capable of detecting when a sample belongs to the target class, even in the absence of labeled data from other classes. The proposed method is applied to a curated dataset containing the chemical properties of 1,615 drug molecules approved by U.S. Food and Drug Administration (FDA), offering a valuable resource for future research. Our findings indicate that integrating geometric and density-based insights improves generalization and risk estimation in one-class learning tasks, providing a robust solution for analyzing noisy, unlabeled data.*

1. Introduction

Identifying meaningful examples in datasets where most or all data belong to a single category is a common and difficult problem in Machine Learning (ML). To deal with this situation, researchers often turn to unsupervised or one-class classification methods [Liu et al. 2008, Breunig et al. 2000, Schölkopf et al. 2001, Ester et al. 1996]. While useful for detecting outliers, these methods typically yield binary decisions and cannot estimate how strongly a new instance belongs to the target class. Additionally, they are sensitive to data sparsity and distributional imbalances, which can degrade performance in low-sample or uneven datasets. Unsupervised methods like clustering group data based on intrinsic similarity, without labels or prior distribution knowledge. As a result, they lack awareness of the target class, often producing multiple clusters even when all data points belong to a single class. In contexts like disease detection, intraclass variability, such as differing symptoms or geographic origins, can cause truly similar instances to be split across clusters or misclassified as outliers. This underscores the limitations of clustering alone in extracting meaningful patterns from single-class data without supervision.

To address the challenges associated with single-class data, we propose a novel approach that combines the strengths of clustering and geometric analysis to uncover layered geometric patterns in noisy, unlabeled data. Our method integrates a clustering technique to identify and eliminate noise with a geometric strategy that reveals the internal structure and organization of the target class. This hybrid approach enables a more precise representation of the shape, boundaries, and diversity within a single class, even in the absence of labeled data from other classes. Rather than relying on data from multiple classes, our approach focuses on exploring the internal structure and boundaries of a single known class. We demonstrate the effectiveness of this methodology through a case study in drug discovery, specifically identifying drug-like molecules. However, the approach is broadly applicable to any problem where only data from one class is available. Our results show that combining geometric and density-based insights can improve generalization and risk estimation in one-class learning scenarios.

The main contributions of this paper are as follows: (i) Development of a novel method for one-class classification, capable of detecting when a sample belongs to the target class; (ii) The creation and public release of a curated dataset containing the chemical properties of 1,615 drug molecules approved by U.S. Food and Drug Administration (FDA), which can serve as a valuable resource for future studies; (iii) The application of the proposed method to this dataset, demonstrating its practical utility in the drug discovery domain and establishing a foundation for further exploration and development of similar approaches in other fields.

2. Theoretical Background

Clustering is a foundational technique in unsupervised and semi-supervised [Bair 2013] learning aiming to partition a dataset into cohesive subsets (clusters) [Xu and Wunsch II 2008]. Ideally, points within the same cluster exhibit high similarity, while points in distinct clusters are more dissimilar.

A fundamental component of clustering algorithms is the use of distance metrics to quantify the similarity between data points. To better understand them, let $S = \{x_1, x_2, ..., x_n\}$ be a dataset consisting of n data points, in which $x_i \in \mathbb{R}^d$, $1 \leq i \leq n$, is a feature vector in a d-dimensional space. Such metrics determine whether two objects, based on their features, should be assigned to the same cluster. A widely used family of distance measures is the Minkowski distance, defined as $d(x_i, x_j) = \sqrt[p]{\sum_{f=1}^{d} |x_i^f - x_j^f|^p}$, which is appropriate for independent and identically distributed observations. Different values of p yield distinct distance measures: for example, $p = 1$, $p = 2$, and $p \to \infty$ correspond to the Manhattan, Euclidean, and Chebyshev (or Supremum) distances, respectively. The choice of p depends on the nature of the data and the desired sensitivity to feature differences. Distance measures play a critical role by directly influencing the partition π produced by clustering algorithms, effectively guiding the formation of the partition $\pi = C_1, C_2, \ldots, C_k$, where each C_k denotes the k-th cluster and $k < n$. The union of all clusters reconstructs the dataset, i.e., $S = \bigcup_{r=1}^{k} C_r$. In the context of this work, the resulting clusters are assumed to satisfy the following properties: (i) $C_r \neq \emptyset$, $\forall C_r \in \pi$; and (ii) $C_r \cap C_q = \emptyset$, $\forall C_r, C_q \in \pi,\ r \neq q$.

Due to the absence of labels, the evaluation of partition quality relies on statistics computed from the clusters and the data itself. According to the literature, evaluation methods are typically categorized into two main branches. The first, known as relative (or internal) criteria, is applied when no ground truth is available. These methods are commonly used to compare clustering results produced either by different algorithms or by the same algorithm under varying parameter settings [Xu and Wunsch II 2008]. In this work, we consider the internal evaluation metric Silhouette Score (SS). Silhouette Score measures how similar a point is to its cluster compared to others [Rousseeuw 1987]. The score ranges from -1 to +1, where a high value indicates that the point is well-matched to its cluster and poorly matched to neighboring clusters, while a negative score indicates that it might be incorrectly assigned. The score $s(x_i)$ for a single data point is defined as $s(x_i) = \frac{\delta(x_i) - \eta(x_i)}{\max\{\eta(x_i), \delta(x_i)\}}$, such that $\eta(x_i)$ is the average distance of x_i to all other points in the same cluster, $\delta(x_i)$ is the minimum average distance of x_i to all points in each of the other clusters. The average Silhouette Score for a partition is computed by $SS(\pi) = \frac{1}{n}\sum_{i=1}^{n} s(x_i)$.

The second evaluation branch focuses on external criteria, which are designed to assess clustering performance based on a predefined structure. In essence, these methods evaluate the degree of correspondence between the estimated partition (π) and a ground truth structure (β) known in advance [Jain et al. 1988]. To better understand these criteria, consider the following cases [Xu and Wunsch II 2008]: **(a)** x_i and x_j belong to the same clusters of π and the same category of β; **(b)** x_i and x_j belong to the same clusters of π but different categories of β; **(c)** x_i and x_j belong to different clusters of π but the same category of β; and **(d)** x_i and x_j belong to different clusters of π and different category of β. Considering these cases, the Fowlkes–Mallows Index (FMI), defined by $\text{FMI}(\pi, \beta) = \sqrt{\frac{a}{a+b} * \frac{a}{a+c}}$, computes a score equivalent to the geometric mean of precision and recall. The index ranges from 0 (no alignment) to 1 (perfect alignment), making it particularly useful for evaluating how well a clustering algorithm recovers the underlying structure of the data when the true partitioning is known.

3. Related Work: Unsupervised Learning

The first related method is Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [Ester et al. 1996], which is a density-based clustering algorithm that groups data points that are closely packed, labeling as noise those in low-density regions. It enables the discovery of arbitrarily shaped clusters without requiring a previously specified number of expected clusters. DBSCAN is particularly useful for datasets containing noise and outliers, making it well-suited for clustering and anomaly detection applications [Toshniwal et al. 2020]. DBSCAN is based on two key parameters: Eps, the maximum distance for two points to be considered neighbors; and $MinPts$, the minimum number of neighbors for a point to be a core point. DBSCAN classifies points as core points, border points, or noise points. Core points have at least $MinPts$ neighbors. Border points are within Eps of a core point but have less than $MinPts$ neighbors. All remaining points become noise points. In summary, DBSCAN is based on 5 steps: (i) Identify the Eps-neighborhood of each point; (ii) Determine which points are core points; (iii) Connect neighboring core points, forming a cluster for each connected set of core points; (iv) Assign border points to the cluster of their closest core point; and (v) Label all remaining points as noise.

Another widely used approach for one-class problems is the One-Class Support Vector Machine (SVM) [Schölkopf et al. 2001]. Unlike density-based methods such as DBSCAN, One-Class SVM is a margin-based model that attempts to learn a decision function that best separates the normal data from the origin, under the assumption that most of the data belongs to a single (normal) class and anomalies are rare and different. The algorithm builds a boundary around the normal data in feature space, maximizing the margin from the origin. As it requires only normal data, One-Class SVM is well-suited for unsupervised settings. Its performance depends on parameters like the kernel function, which balances outlier tolerance and model complexity.

An efficient approach to addressing one-class problems in moderately high-dimensional datasets is the use of the Local Outlier Factor (LOF) algorithm [Breunig et al. 2000]. LOF offers an efficient solution for one-class problems in moderately high-dimensional data by assessing abnormality based on local density. Unlike global models, LOF compares a point's density to that of its K-nearest neighbors, flagging points in sparser regions as outliers. This local perspective makes it well-suited for datasets with heterogeneous densities or internal clustering, where global methods may fail. Finally, the Isolation Forest (IF) [Liu et al. 2008] is also a well-known method for unsupervised one-class problems. Based on the idea that anomalies are few and different, it builds an ensemble of random trees to isolate data points. Anomalies, being easier to separate, yield shorter average path lengths. In one-class settings, IF learns isolation patterns from normal data alone, flagging deviations without needing prior knowledge of anomalies. Despite their effectiveness in various anomaly detection scenarios, the aforementioned methods exhibit some limitations when applied to one-class problems. These include sensitivity to parameter selection, difficulty handling high-dimensional or noisy data, and challenges in capturing the internal structure of the normal class when no labeled anomalies are available. Furthermore, many of these approaches either assume uniform data distributions or struggle to distinguish between noise and true anomalies, often leading to degraded performance in complex or irregular datasets. To address these challenges, the following section introduces a novel method that integrates a clustering technique to identify and eliminate noise, coupled with a geometric strategy designed to uncover the intrinsic structure and organization of the target class. This approach aims to enhance robustness in unsupervised one-class learning tasks.

4. Proposed Approach

The proposed approach was designed to deal with data with a single known label. It is important to highlight that our proposal is devoted to not only finding clusters in data, but also to indicating the degree of membership of an instance to a single class, based on its relative depth within the group structure. Unlike probabilistic approaches, this method evaluates the possibility of an instance belonging to the class by assessing its embeddedness in the cluster, considering factors such as density and proximity to core points. The deeper an instance lies within the cluster's structure, the higher its associated membership, reflecting a stronger affiliation with the group. This approach is particularly useful for situations where traditional probability-based confidence scores are not applicable. To reach this goal, our approach was designed on top of two main methods. Initially, it runs DBSCAN to find clusters in the analyzed data. We chose

DBSCAN over the alternative methods described in Section 3 due to its ability to detect clusters of arbitrary shapes and its robustness to noise. Secondly, we incorporate the Convex Hull Onion Peeling method, a widely studied approach based on computational geometry, to estimate the depth of an analyzed instance within a given cluster.

The convex hull of a set of points S can be intuitively defined as the intersection of all convex sets that contain S, or the intersection of all half-spaces that contain S [O'Rourke 1998]. A point x_i in the convex hull of a set S is a boundary point if at least one supporting hyperplane of the convex hull exists, such that x_i lies on the hyperplane itself. In convex hull onion peeling, a convex hull algorithm is applied iteratively to a set of points. In each iteration, the boundary points of the current hull are removed before computing the next hull. This process continues until fewer than $d+1$ affinely independent points remain, where d is the dimensionality of the dataset (since $d+1$ points are required to form a d-simplex). This results in a partial ordering where each point belongs to a numbered layer, and points in the outermost layers are likelier to be outliers. Each computed convex hull is stored as a collection of half-space inequalities. Each half-space boundary uniquely contains one of the convex hull's *d-1–faces*. The union of the half-spaces is the d-dimensional polytope bounded by the convex hull. This representation is computationally optimal for determining whether a point is inside a convex hull and, thus, within which layer of a cluster it belongs. In summary, the proposed method comprises four steps, which can be further subdivided into training (steps 1 and 2) and inference (steps 3 and 4): 1) apply density-based clustering on the training set; 2) apply convex hull onion peeling to compute nested convex hulls for each cluster; 3) For any unseen test sample x_{new}, determine the cluster $C_r \in \pi$ to which it would belong; 4) determine the innermost convex hull layer that encloses the instance, and calculate a membership value based on the ratio of points in exterior versus interior layers.

The membership value of a new point x_{new} is computed by sequentially traversing the precomputed convex hull layers of the assigned cluster C_r, evaluating point inclusion through the half-space inequalities that define each layer. Once the innermost containing layer is identified, the membership is calculated as the ratio between the number of points in the containing and outer layers (n_{external}) and the total number of points in the cluster ($|C_r|$): $\mathcal{M}(x_{\text{new}}) = \frac{n_{\text{external}}}{|C_r|}$ if $x_{\text{new}} \in C_r$ and $C_r \in \pi$.

To illustrate our approach, consider the example shown in Figure 1, where dots represent data points in $\mathbb{R}^d$. After applying DBSCAN (Step 1), the points are either assigned to clusters, distinguished as large (blue) or small (green), or identified as noise (red). Dashed lines represent the boundaries of the convex hulls computed in Step 2. Data points intersected by these boundaries belong to the corresponding convex hull layer. Hollow squares indicate new instances being evaluated by the method.

The red square represents a new instance classified as noise in Step 3. The dark blue square was assigned to a cluster in Step 3 but lies outside all convex hull layers, resulting in a membership value of $\mathcal{M}(\cdot) = 0$. Light blue squares correspond to instances assigned to clusters and located within one or more convex hull layers of their respective clusters, thus receiving higher membership values (Step 4). The execution of Step 3 is particularly challenging due to the task of assigning new instances to existing clusters. To clarify this process, Algorithm 1 outlines the procedure used to perform this

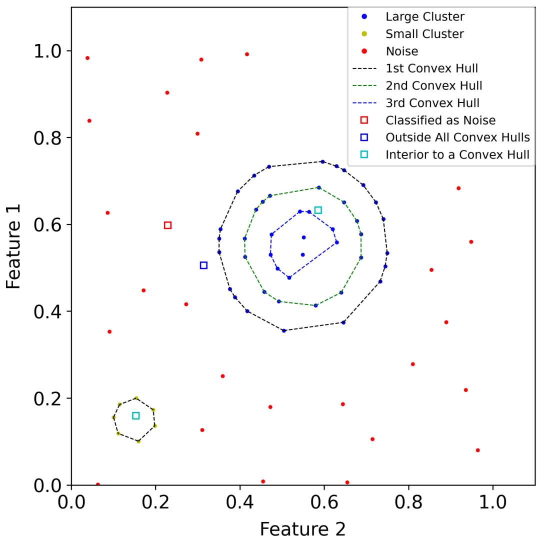


Figure 1. Example illustrating the execution of the proposed approach.

assignment effectively.

To determine the cluster to which a new test instance x_{new} would belong, we simulate how DBSCAN would assign it, as shown in Lines 1-10. The assignment can be efficiently computed using only the point coordinates and existing cluster labels. First, we compute two distance-based neighborhood sets: (i) N_{Eps}: Eps-neighborhood of x_{new}; and (ii) N_{2Eps}: points with distance to $x_{new} \leq 2 \times Eps$.

In Lines 12-23, the algorithm identifies core points within N_{Eps} by checking for density reachability. For each point $x \in N_{Eps}$, we determine if x_{new} has at least $MinPts$ neighbors within Eps distance by examining all $x_i \in N_{2Eps}$. Identifying core points is executed in $\mathcal{O}(dn^2)$ in the worst case, where d is the dimensionality and n the number of points in N_{2Eps}, which can be equal to the size of the training dataset. Its cost can be amortized across many inferences by precomputing the set of core points in the training dataset. As shown in Lines 25-31, if no core points are found in N_{Eps}, x_{new} is labeled as noise. Otherwise, x_{new} is assigned to the cluster of the nearest core point.

Identifying the nearest core point is $\mathcal{O}(dn)$ in the worst case. The overall time complexity of this step of the proposed method is $\mathcal{O}(dn^2)$, or $\mathcal{O}(dn)$ if a set of core points is available. While a more straightforward approach of assigning x_{new} to the cluster of its nearest neighbor when their distance is less than Eps is possible, this may lead to imprecise results. Although points farther than Eps are necessarily noise, the converse does not hold: proximity within Eps is insufficient to guarantee cluster membership without considering density connectivity.

5. Experimental Setup

Experiments were conducted to evaluate the proposed method using a dataset comprising 1,615 FDA-approved drug molecules. Each molecule in the dataset is characterized by 11 distinct features representing various chemical properties. After eliminating duplicates, the total number of molecules was reduced to 1,225. Given the sensitivity of clustering algorithms employing Minkowski distance metrics (e.g., Euclidean and

Algorithm 1 Assigning a Test Sample to a Cluster.

```
Require: Points $S$, Clusters $\pi$, New Point $x_{new}$, $Eps$, $MinPts$
Ensure: Updated Clusters $\pi'$
$N_{Eps} \leftarrow \emptyset$
$N_{2Eps} \leftarrow \emptyset$
for each $x \in S$ do
    if $dist(x, x_{new}) \leq Eps$ then
        $N_{Eps} \leftarrow N_{Eps} \cup \{x\}$
        $N_{2Eps} \leftarrow N_{2Eps} \cup \{x\}$
    else if $dist(x, x_{new}) \leq 2 * Eps$ then
        $N_{2Eps} \leftarrow N_{2Eps} \cup \{x\}$
    end if
end for

$core\_points \leftarrow \emptyset$
for each $x \in N_{Eps}$ do
    $neighbor\_count \leftarrow 0$
    for each $y \in N_{Eps} \cup N_{2Eps}$ do
        if $dist(x, y) \leq Eps$ then
            $neighbor\_count \leftarrow neighbor\_count + 1$
        end if
    end for
    if $neighbor\_count \geq MinPts$ then
        $core\_points \leftarrow core\_points \cup \{x\}$
    end if
end for

if $core\_points = \emptyset$ then
    $\pi' \leftarrow \pi \cup \{(x_{new}, noise)\}$
else
    $x_{closest} \leftarrow \text{argmin}_{x \in core\_points} dist(x, x_{new})$
    $cluster \leftarrow \pi[x_{closest}]$
    $\pi' \leftarrow \pi \cup \{(x_{new}, cluster)\}$
end if

return $\pi'$
```

Manhattan distances) to feature scale variations, Min-Max normalization was applied to constrain all numerical features to the interval [0,1]. This ensures equal contribution of each feature to the clustering analysis while preserving internal relationships.

Another important preprocessing step is the analysis of dimensionality reduction. However, it is important to emphasize that, in any clustering-based approach, removing features can modify the shape and structure of clusters in the high-dimensional space. As observed in Figure 2, as the dimensionality increases, the number of layers from the convex hull onion peeling on remaining points not selected in prior iterations tends to decrease. The presence of multiple convex layers is important, as it enables a higher level of granularity in classification. A sample in deeper layers indicates a stronger alignment with the original cluster. For example, considering dimensionalities from 2 to 7, the distribution of samples across convex layers exhibited characteristics consistent with a normal distribution. The sample count demonstrated a monotonic increase through successive convex layers until reaching a maximum, followed by a constant decline. This distributional pattern could not be observed in cases where dimensionality exceeded 8, owing to the small number of resultant layers.

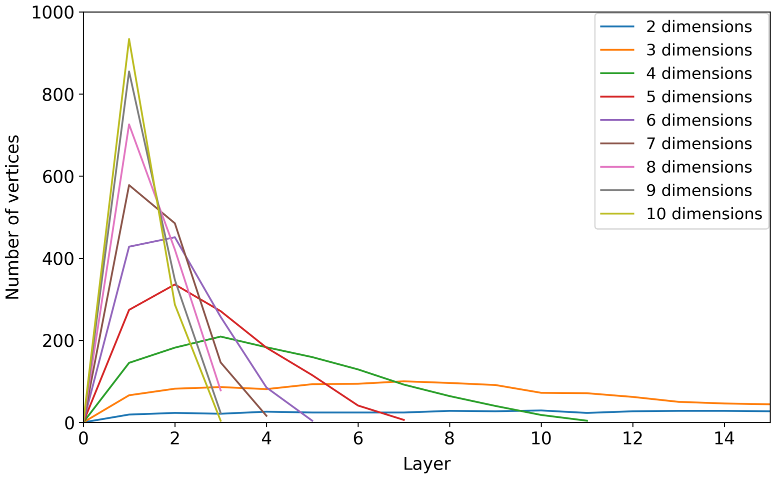
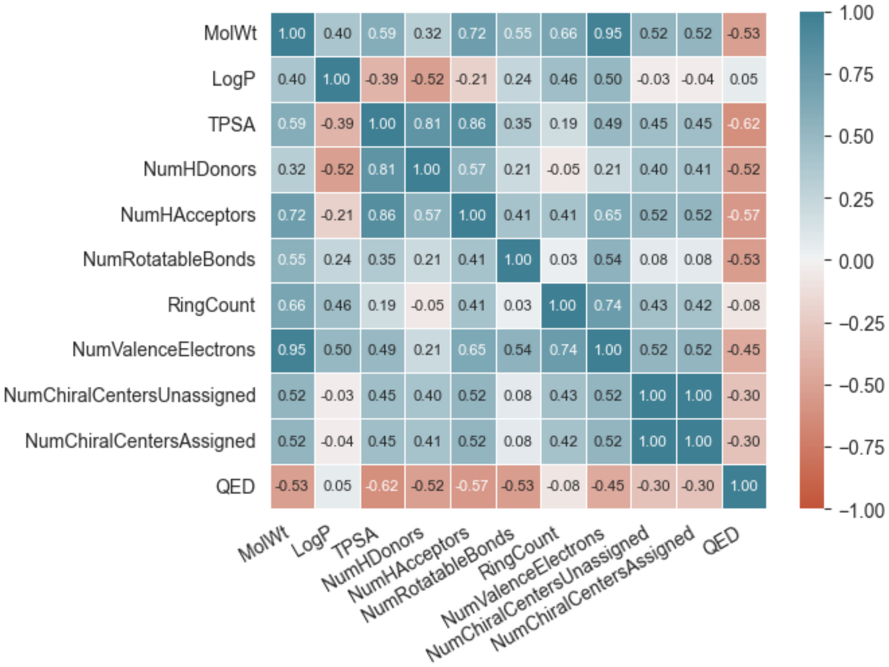


Figure 2. Convex Hull size per layer for each dimensionality.

We used the Pearson Correlation Coefficient (PCC) to measure the relationship between our features (Figure 3) The features `NumChiralCenterAssigned` and `NumChiralCentersUnassigned` exhibited an exceptionally high PCC of 1.00. Given their semantic similarity, they were combined into a new feature, `NumChiralCenters`, which was subsequently Min-Max normalized to ensure a consistent scale. Although other features are highly correlated, no further dimensionality reduction was applied to minimize the risk of losing relevant information.



	MolWt	LogP	TPSA	NumHDonors	NumHAcceptors	NumRotatableBonds	RingCount	NumValenceElectrons	NumChiralCentersUnassigned	NumChiralCentersAssigned	QED
MolWt	1.00	0.40	0.59	0.32	0.72	0.55	0.66	0.95	0.52	0.52	-0.53
LogP	0.40	1.00	-0.39	-0.52	-0.21	0.24	0.46	0.50	-0.03	-0.04	0.05
TPSA	0.59	-0.39	1.00	0.81	0.86	0.35	0.19	0.49	0.45	0.45	-0.62
NumHDonors	0.32	-0.52	0.81	1.00	0.57	0.21	-0.05	0.21	0.40	0.41	-0.52
NumHAcceptors	0.72	-0.21	0.86	0.57	1.00	0.41	0.41	0.65	0.52	0.52	-0.57
NumRotatableBonds	0.55	0.24	0.35	0.21	0.41	1.00	0.03	0.54	0.08	0.08	-0.53
RingCount	0.66	0.46	0.19	-0.05	0.41	0.03	1.00	0.74	0.43	0.42	-0.08
NumValenceElectrons	0.95	0.50	0.49	0.21	0.65	0.54	0.74	1.00	0.52	0.52	-0.45
NumChiralCentersUnassigned	0.52	-0.03	0.45	0.40	0.52	0.08	0.43	0.52	1.00	1.00	-0.30
NumChiralCentersAssigned	0.52	-0.04	0.45	0.41	0.52	0.08	0.42	0.52	1.00	1.00	-0.30
QED	-0.53	0.05	-0.62	-0.52	-0.57	-0.53	-0.08	-0.45	-0.30	-0.30	1.00

Figure 3. Correlation matrix (Pearson Correlation Coefficient).

To assess the proposed method's effectiveness, a randomized dataset was produced by shuffling the values of each feature independently, ensuring only valid values observed in the original dataset were used. This randomization process yielded a suitable synthetic dataset for evaluating the method's performance in rejecting noise, while also enabling the use of both internal (Silhouette and Fowlkes-Mallows) and external (Precision, Recall, and F1-score) clustering validation metrics to comprehensively assess the quality of the grouping and class membership assignment. The resulting synthetic dataset contained 1,615 unique samples, which were then Min-Max normalized using the same parameters as the original dataset and subjected to the same dimensionality reduction step.

6. Results

Our experiments started by optimizing the DBSCAN's hyperparameters through a full factorial grid search over the parameter space, evaluated on the complete original dataset. The search explored 97,760 combinations of values for $Eps = [0.01, 0.02, \ldots, 1.6]$ and $MinPts = [2, 3, \cdots, 612]$. The optimal hyperparameters were chosen based on silhouette, with the additional constraint of requiring noise to account for 1% to 20% of the samples. The best score was 0.6360 for the hyperparameters $Eps = 0.39, MinPts = 4$. The ten highest scores obtained are presented in Table 1.

Table 1. Metrics from DBSCAN Hyperparameter Grid Search.

Eps	MinPts	Silhouette Score[1]	Fowlkes-Mallows Index[2]	Clusters	Noise
0.39	4	0.6360	0.6831	18,1202	5
0.39	5	0.6360	0.6817	18,1202	5
0.40	4	0.6342	0.6868	16,1204	5
0.40	5	0.6342	0.6851	16,1204	5
0.41	6	0.6184	0.6887	15,1204	6
0.42	6	0.6174	0.6917	14,1205	6
0.42	5	0.6163	0.6925	13,1206	6
0.41	5	0.6163	0.6902	13,1206	6

[1] The Silhouette score was computed only for points assigned to clusters, excluding points labeled as noise by DBSCAN.

[2] The Fowlkes-Mallows index was calculated by combining the original and synthetic (randomized) datasets. The true labels were established by treating each dataset as a distinct cluster. Following DBSCAN clustering on the combined dataset, noise points were assigned to the synthetic dataset's cluster, while all other clustered points were consolidated into a single cluster representing the original dataset.

Analyzing DBSCAN, as shown in Table 2, it is possible to notice that it was not capable of effectively separating the original dataset from the synthetic (noise-like) dataset. DBSCAN only rejected 26.98% of the synthetic dataset's samples, successfully labeling them as noise. An overwhelming majority of the samples (73.02%) were assigned to the largest, and more spatially dispersed, cluster.

By considering our proposal, one may noticed that the results (Table 3) improved significantly after executing the convex hull onion peeling step. Although most

Table 2. Molecule incidences computed only using DBSCAN.

Dataset	Large Cluster	Small Cluster	Noise
Original	98.12%	0.41%	1.47%
Synthetic	73.02%	0.00%	26.98%

samples were assigned to the largest cluster, convex hull onion peeling classified most of the synthetic dataset (72.52%, or 99.31% of the samples assigned to the large cluster) as external to all of the cluster's layers and thus unlikely to belong to it.

Table 3. Molecule incidences - DBSCAN followed by Convex Hull Onion Peeling.

Group[1]	Original Dataset[2]	Synthetic Dataset[3]
Noise	1.47%	26.98%
Large Cluster (external)	-	72.52%
Large Cluster (at 1st layer)	79.35%	-
Large Cluster (between 1st and 2nd layer)	-	0.43%
Large Cluster (at 2nd layer)	18.53%	-
Large Cluster (within 2nd layer)	0.24%	0.06%
Small Cluster	0.41%	0.00%

[1] Layers are numbered from the most external to the most internal.

[2] As a percentage of the entire original dataset.

[3] As a percentage of the entire synthetic dataset.

A small fraction of the dataset (0.49%, or 0.67% of the samples assigned to the large cluster) was classified as within one of the internal layers. As the first convex hull contains most molecules of the cluster in the original dataset, the membership probability assigned to these samples is relatively high: 80.87% for the samples located between the first and second hull and 99.75% for samples internal to the second hull. Table 4 summarizes the results: 99.51% of the synthetic dataset samples were rejected correctly. In other words, our method achieved an accuracy of 0.9951. To highlight the significance of this result, we repeated the experiment using classical approaches. One-Class SVM classified all observations as noise, while LOF and IF achieved performances of 66.21% and 39.11%, respectively. Although our approach demonstrated superior performance, it is important to note that we did not conduct an in-depth analysis of these methods, as they were originally designed for different objectives.

Table 4. Molecule incidences - DBSCAN + Convex Hull Onion Peeling (simplified).

Dataset	Within a cluster	Outside any cluster
Original	98.53%	1.47%
Synthetic	0.49%	99.51%

In our final experiments, we employed classification metrics to assess the likelihood of correctly identifying inliers (i.e., molecules located within clusters), as shown in Table 5, using a 10-fold cross-validation strategy. In a conservative scenario, lower membership values would lead to the non-recommendation of drugs located in small dense groups or near cluster borders. Based on Table 5, it is evident that the highest performance was achieved in terms of Precision, that is, no false positives (FP) were produced by our method. On the other hand, the Recall metric reflects the conservative nature of the approach, as some observations with lower membership values were not recommended, which is apparent from the true positive (TP) and false negative (FN) rates. Finally, the F1-Score summarizes this trade-off, demonstrating that even under a conservative strategy, our method delivers relevant and reliable results.

Table 5. Results of 10-fold cross-validation on inliers (molecules inside clusters).

Fold	Precision	Recall	F1-Score	TP Rate	FP Rate	FN Rate
0	1.0000	0.5455	0.7059	0.5455	0.0000	0.4545
1	1.0000	0.7273	0.8421	0.7273	0.0000	0.2727
2	1.0000	0.5417	0.7027	0.5417	0.0000	0.4583
3	1.0000	0.2500	0.4000	0.2500	0.0000	0.7500
4	1.0000	0.4583	0.6285	0.4583	0.0000	0.5417
5	1.0000	0.8696	0.9303	0.8696	0.0000	0.1304
6	1.0000	0.3478	0.5161	0.3478	0.0000	0.6522
7	1.0000	0.4783	0.6471	0.4783	0.0000	0.5217
8	1.0000	0.5652	0.7222	0.5652	0.0000	0.4348
9	1.0000	0.5909	0.7429	0.5909	0.0000	0.4091
Average	1.0000	0.5375	0.6838	0.5375	0.0000	0.4625

About the temporal performance, DBSCAN has computational complexities equal to $\mathcal{O}(n^2)$. The convex hull algorithm has time complexity given by $O(nf_r/r)$, where n represents the total number of points in the dataset, r denotes the number of points in the resulting convex hull, and f_r is the maximum number of facets for a polytope with r vertices. Although these methods exhibit limited performance in problems with a large number of features, they are still considered state-of-the-art. If faster alternatives become available, either method can be replaced within our approach.

7. Final Remarks

The experiments described in this paper effectively demonstrated the feasibility and effectiveness of combining noise-identifying clustering methods (DBSCAN) with convex hull onion peeling to tackle one-class classification problems. The proposed approach showed a notable enhancement (26.98% vs. 99.51%) in distinguishing between the FDA-approved drug molecules (original dataset) and the randomized samples (synthetic dataset) compared to DBSCAN alone. This demonstrates a higher specificity for potentially relevant molecules, offering a valuable tool for the early stages of drug discovery by helping to prioritize research efforts and investments in molecules with a higher likelihood of becoming FDA-approved drugs. Although our approach demonstrates strong potential in modeling single-class data, certain limitations may impact its scalability in large-scale applications. In particular, the computation of convex hulls can incur significant memory and time costs. However, this issue can be effectively addressed through the adoption of more efficient convex hull algorithms [Goodrich and Kitagawa 2024, Siegel 2022, Leng et al. 2019] and the use of parallel processing techniques, which open avenues for scaling the method to larger and more complex datasets without compromising its geometric interpretability.

Data and Code Availability: The datasets and source codes used in this study are available at `https://github.com/LabIA-UFBA/dbscan-onion-peeling`.

Acknowledgements

This work was supported by CNPq (Brazilian National Council for Scientific and Technological Development), Maria Emilia Foundation, FAPESB (Bahia Research Foundation), and CAPES (Coordination for the Improvement of Higher Education Personnel – Brazil).

References

Bair, E. (2013). Semi-supervised clustering methods. *WIREs Computational Statistics*, 5(5):349–361.

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

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. KDD'96, page 226–231. AAAI Press.

Goodrich, M. T. and Kitagawa, R. (2024). Making quickhull more like quicksort: A simple randomized output-sensitive convex hull algorithm.

Jain, A. K., Dubes, R. C., et al. (1988). *Algorithms for clustering data*, volume 6. Prentice hall Englewood Cliffs.

Leng, Q., Wang, S., Qin, Y., and Li, Y. (2019). An effective method to determine whether a point is within a convex hull and its generalized convex polyhedron classifier. *Information Sciences*, 504:435–448.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE.

O'Rourke, J. (1998). *Computational geometry in C.* Cambridge University Press, Cambridge, United Kingdom.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.

Siegel, A. (2022). A parallel algorithm for understanding design spaces and performing convex hull computations. *Journal of Computational Mathematics and Data Science*, 2:100021.

Toshniwal, A., Mahesh, K., and Jayashree, R. (2020). Overview of anomaly detection techniques in machine learning. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 808–815.

Xu, R. and Wunsch II, D. C. (2008). *Clustering.* IEEE Press Series on Computational Intelligence. Wiley, Hoboken, NJ, 1 edition.