Optimized DBSCAN Parameter Selection: Stratified Sampling for Epsilon and GridSearch for Minimum Samples

Gloriana Joseph Monko and Masaomi Kimura

Department of Functional Control Systems and Department of Computer Science and Engineering, Shibaura Institute of Technology, Tokyo, Japan

Abstract. This research presents an advanced methodology for estimating the epsilon and minimum samples parameters in the DBSCAN clustering algorithm using a Stratified Sampling and Grid-Search approach. Our method showcased notable improvement in eps estimation precision across nine diverse datasets compared to conventional techniques. By accounting for dataset variations in structure and density, stratified sampling leads to superior cluster formations. The k-nearest distance graph further refines these relationships, ensuring a comprehensive understanding of data densities. Additionally, our method underscores the importance of each dataset's unique stratum, providing holistic insights. We also introduced a Grid-Search technique for MinPts estimation with the help of silhouette score, challenging traditional rule-of-thumb settings. Our approach suggests setting MinPts flexibly, considering the dataset's specific attributes and has proven its efficacy by enhancing clustering results, with implications for both SS-DBSCAN and traditional DBSCAN frameworks. This study highlights the potential of parameter estimation in optimizing clustering outcomes and computational efficiency.

Keywords: epsilon selection, MinPts determination, statified sampling, grid-search, SS-DBSCAN.

1 Introduction

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [1] is an algorithm that has revolutionized the domain encompassing data mining and machine learning over the past few decades [1]. Unlike many traditional clustering techniques that rely on spherical assumptions or require explicit knowledge about the number of clusters, DBSCAN relies on the data's inherent structure [2], [3]. The algorithm functions by detecting clusters through an assessment of data point density, making it adept at detecting clusters of varying shapes and sizes and differentiating between cluster points and noise. As DBSCAN identifies clusters based on the density of data points, it naturally positions clusters within high-density regions, while outliers tend to reside in low-density areas, as illustrated in Fig. 1.

However, the effectiveness of DBSCAN is intrinsically tied to its parameters: Epsilon ϵ and Minimum Samples (MinPts). While ϵ defines the radius to search for neighboring points [4], MinPts specifies the minimum number of points required to form a dense region [4]. Although the choice of ϵ has received considerable attention in the literature [5], [6], [7], [8], [9], [10] and various techniques have been proposed for its determination, the selection of MinPts has been somewhat overlooked. The 'one-size-fits-all' approach [11], where a generic value of MinPts is used across various datasets, can lead to suboptimal clustering results. The intricacy lies in the fact that the ideal value of MinPts relies on the dataset size and its inherent structure and distribution.

DBSCAN's parameters, ϵ and MinPts, are foundational in its operation [12],[13],[14]. While the significance of ϵ in delineating the neighborhood radius is well understood, its determination has often posed challenges, leading to a plethora of research in that direction.

David C. Wyld et al. (Eds): SEAU, NLMLT, ITCS, CST, DMS -2023
pp. 43-61, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.132004

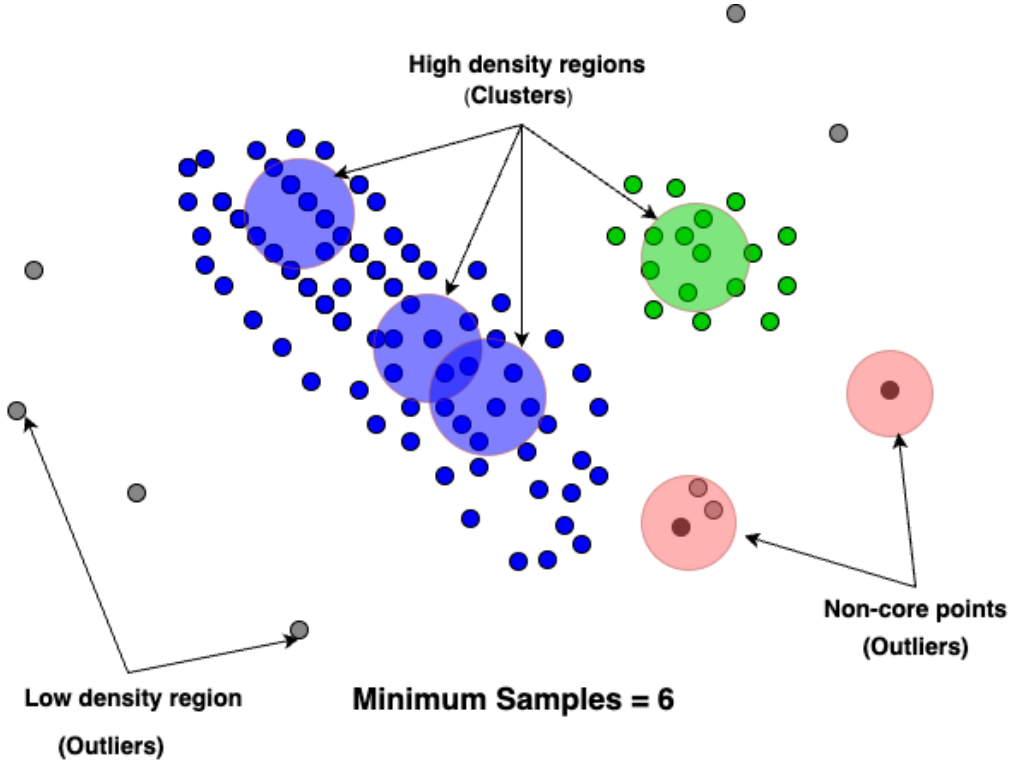

Fig. 1: DBSCAN Clusters

In our previously accepted work by Gloriana Monko, Masaomi Kimura, "SS-DBSCAN: Epsilon Estimation with Stratified Sampling for Density-Based Spatial Clustering of Applications with Noise", (to be published), we addressed this challenge by introducing a novel approach for ϵ parameter estimation using Stratified Sampling. The methodology was grounded in the idea that by partitioning the data into more homogeneous subsets, we could more effectively estimate the optimal ϵ value for each subset, leading to enhanced clustering results on the consolidated data. Having delved into the intricacies of ϵ estimation, we now focus on the equally crucial but often overlooked MinPts parameter. This exploration into MinPts determination using the Grid-Search technique is a natural progression and extension of our prior work by Gloriana Monko, Masaomi Kimura, "SS-DBSCAN: Epsilon Estimation with StratifiedSampling for Density-Based Spatial Clustering ofApplications with Noise",(to be published), further enhancing the robustness and efficacy of the SS-DBSCAN algorithm.

This paper ventures into validating the ϵ selection using stratified sampling techniques discussed in our previous work and MinPts determination for SS-DBSCAN. Recognizing the pivotal role of MinPts in defining the granularity of clusters and its impact on the algorithm's sensitivity to noise, we introduce a comprehensive methodology for its optimization using the Grid-Search technique. Traditionally employed for hyperparameter tuning in various machine learning algorithms [15], [16], Grid-Search tests a predefined set of values and evaluates their performance to pinpoint the optimal choice. By extending the application of Grid-Search to the context of DBSCAN's MinPts determination, this research seeks to bridge an existing gap in the clustering domain.

As we progress through this paper, we will acquaint readers with sections concerning prior research, the theoretical foundations of our approach, and a comprehensive presentation of our experiments, outcomes, and discussion.

2 Related Works

The quest for optimized clustering, particularly in the context of the DBSCAN algorithm and the automatic selection of its parameters, has been the focus of numerous studies in the field of data mining and machine learning. This section reviews some of the pivotal works in this domain, shedding light on the evolution of techniques and the current state of research.

2.1 Eps Selection

Ester et al. [1] laid the foundation for density-based clustering by introducing DBSCAN. Their work expounded the algorithm's proficiency in identifying arbitrarily shaped clusters and emphasized the significance of the eps parameter, suggesting k-distance graphs as a heuristic tool for its determination. The concept of k-distance graphs was further expanded by Schubert et al. [17]. Their study delved deeper into the methodology, highlighting its limitations and suggesting enhancements for more accurate "elbow" detection. Starczewski et al.[12] introduced a novel approach to identify pronounced distance spikes by leveraging the kdist function. This function computes the distances between data points and their respective kth nearest neighbors within a dataset. Their methodology offers a fresh perspective on discerning patterns and anomalies in spatial relationships, emphasizing the importance of understanding the nuanced interactions between data points and their immediate surroundings. Advancements in optimizing DBSCAN took a significant step forward with the work of Lai et al. [5]. They pioneered an approach grounded in the 'MVO-multiverse optimizer algorithm', focusing on the iterative enhancement of DBSCAN parameters. Their technique provided a dynamic means of achieving the best clustering results. In a related development, Khan et al. [18] brought forth an adaptive variant of DBSCAN, appropriately termed adaptive DBSCAN. Their strategy was crafted with precision to automate the selection of ideal parameter values, including epsilon and the minimum number of points. Dawid and Krzysztof [19] presented GrDBSCAN, which partitions data into fuzzy granular representations and subsequently applies density-based clustering to these derived granules. Their method mainly focused on reducing clustering execution time.

2.2 MinPts Determination

Another DBSCAN parameter estimation technique considers the determination of the minimum points parameter. This parameter defines the minimum data points requirement for establishing a dense region Fig. 1. Schubert et al. [17] suggests an easier way to set MinPts parameter of DBSCAN by using a default value of 4. It's purpose is to refine the density estimate, and for the large datasets, the default setting for MinPts in two-dimensional data is 4 [1]. On the other hand, [7] suggests setting it to twice the dataset dimensionality, i.e., MinPts = 2*dim. In cases involving noisy, extensive, high-dimensional datasets, or those with numerous duplicates, increasing the MinPts value might lead to improved outcomes. Concerning MinPts, Breunig et al. [20] proposed a method to find an optimal MinPts value based on the dimensionality of the dataset. They argued that as the dimensionality of a dataset increases, a larger MinPts value is needed to avoid the curse of dimensionality. Nevertheless, this strategy may not consistently produce the most favorable outcomes, particularly in the case of datasets with lower dimensions or varying densities.

3 Proposed Methodology

Our proposed methodology offers a straightforward approach to identifying the optimal eps value using stratified sampling and MinPts value through grid-search. This approach accommodates the unique characteristics of diverse datasets, enabling us to evaluate our algorithm's robustness and versatility across a range of data properties.

3.1 Dataset

In our research, we conducted experiments using two distinct datasets: synthetic datasets and real-world datasets retrieved from GitHub. These real-world datasets (Iris, Iono, Sonar, and Arrhythmia) are inherently complex and characterized by many features as 5, 35, 61, and 263, respectively. We applied dimensionality reduction techniques to handle such high-dimensional data and prepare it for clustering. Specifically, for the real-world datasets, we employed Principal Component Analysis (PCA) and t-Stochastic Neighbor Embedding (t-SNE) to transform the original data into a more manageable and informative representation, resulting in a reduced feature space consisting of two essential features.

In contrast, the synthetic dataset (2d-20c- no0, elly 2d10c13s, sizes1, square4, and st900) used in our study was presented in a numerical format and featured only two distinct attributes. This simplicity facilitated the direct application of our method without the need for dimensionality reduction. By leveraging dimensionality reduction techniques for real-world datasets, we aimed to mitigate the challenges posed by the curse of dimensionality while preserving the relevant information necessary for meaningful clustering. This approach allowed us to ensure the effectiveness of our proposed algorithm across a spectrum of dataset complexities and dimensions.

3.2 Use of K-neighbors

Our study adopted a K-neighbors approach to determine the optimal values for eps and MinPts parameters. These two steps are intertwined, such that the outcome of the first influences the second. The underlying principle is that for a data point to be considered part of a cluster, it must have a minimum number of neighboring points within a specified radius, denoted as eps. Therefore, we established a minimum threshold for the number of neighbors k and computed the distance of each data point to its nearest neighbor. We considered the size of the dataset under investigation to determine the appropriate number of neighbors. Subsequently, we calculated the average distance from each data point to its nearest neighbors. We then applied stratified sampling to the distances derived from this average distance measurement. This process was finalized by creating a k-distance graph based on the stratified sampled distances. To identify the optimal epsilon value, we employed the knee locator technique [21] in conjunction with the elbow method [22] . This method allows us to pinpoint a specific point on the curve where the rate of change in the distances shifts significantly, indicating the appropriate epsilon value. Our emphasis on the average distance captures neighboring points effectively, providing a more comprehensive view of the local data structure, especially in scenarios where outliers are present; we described this process well in (our previously accepted paper). By dividing the average distances into strata and sampling representative distances, this process provides insights into the underlying data structure while ensuring efficient use of computational resources.

3.3 Agorithms for Epsilon Estimation and MinPts Determination

Algorithm 1 and Algorithm 2 delineate the systematic procedure employed to ascertain the values of both pivotal parameters, namely, eps and MinPts.

Algorithm 1 Stratified Sampling for Epsilon Estimation

Input:
Sample data X, where $X = \mathbb{R}^d$
Number of nearest neighbors (k)
Number of strata p
Number of elements sampled from each stratum m
Output:
Optimal epsilon ϵ value for constructing a k-nearest distance graph
Initialization:
Initialize distances array: distances[]
Step 1: Compute Distances
for each element x in X **do**
  for each other_element x' in X **do**
    if x is not equal to x' **then**
      Calculate distance using Euclidean formula: $\mu = \sqrt{\sum_{i=1}^{d}(x_i - x_i')^2}$
      Append μ to distances[]
    end if
  end for
end for
Step 2: Order Distances
Sort distances[] in ascending order
Step 3: Calculate Average Distances
Initialize averages array: averages[]
for each element x in X **do**
  Get the first k distances from distances[] (excluding 0 distance)
  Calculate average distance: $\pi_{\text{avg}}\mu_d = \frac{1}{k}\sum_{i=1}^{k}\pi_i\mu_d$
  Append $\pi_{\text{avg}}\mu_d$ to averages[]
end for
Step 4: Stratified Sampling
Initialize strata array: strata[]
for each $\pi_{\text{avg}}\mu_d$ in averages[] **do**
  Divide $\pi_{\text{avg}}\mu_d$ into p class intervals of equal size: $\tau_p = \pi_{\text{avg}_{\text{lower-limit}}}\mu_d - \pi_{\text{avg}_{\text{upper-limit}}}\mu_d$
  Sample m distances from each stratum using $sample(\tau_p, m)$
  Append sampled distances to sampled_distances[]
end for
Step 5: Construct k-Nearest Distance Graph
Generate a k-nearest distance graph from sampled_distances[]
Locate optimal ϵ value using knee locator technique
Output:
Set $d_{\min}$ (optimal value) as ϵ

Algorithm 2 Determine Optimal MinPts

distancesStrata: A list of distance values for stratified sampling
kneeStrata: A knee locator object used to find the optimal epsilon (eps) value
$features$: The dataset for which DBSCAN clustering is performed
Print the average silhouette scores for different MinPts values
for i in range(3, n, 1) **do**
  Set $epsValue = distancesStrata[kneeStrata.knee]$
  Set $minSamples = i$
  Perform SS-DBSCAN clustering on $features$ with parameters:
    $eps = epsValue$
    $MinPts = minSamples$
  Calculate the silhouette score for the clustering result:
    $silhouetteAvg = silhouetteScore(features, db.labels)$
  Print the results:
    "For min sample value = " + str(minSamples)
    "The average silhouetteScore is : ", silhouetteAvg
end for

3.4 Grid-search for MinPts determination

While there are suggestions and heuristics for choosing MinPts, there is no universally accepted method for its estimation. The best approach depends on the dataset's specific properties and the analysis context. For estimating the MinPts value, we use the grid-search approach. This method ties the MinPts value to the point density within an eps radius, making the MinPts value reliant on the initially set number of neighbors. We recommend setting a lower value for smaller datasets, whereas larger, more complex datasets benefit from a higher one. Once the eps value is determined, MinPts can be deduced based on the following criteria:

- For datasets of 3000 samples or fewer, the range for MinPts should be set between 3 and 100, iterating by 1 or n steps while maintaining the eps value derived from the k-distance graph. The most suitable MinPts value can be selected based on the highest Silhouette score.
- For datasets exceeding 3000 samples, the range for MinPts should commence from 3 or any number up to a designated maximum number, iterating by n steps. Again, the eps value extracted from the k-distance graph remains constant. The optimal MinPts value can be chosen based on the Silhouette score's pinnacle Fig. 2.

```
For min sample value = 3 The average silhouette_score is :  0.5607273501855157
For min sample value = 4 The average silhouette_score is :  0.5603010793851531
For min sample value = 5 The average silhouette_score is :  0.5597740720951461
For min sample value = 6 The average silhouette_score is :  0.5428457878144779
For min sample value = 7 The average silhouette_score is :  0.63193349007879
For min sample value = 8 The average silhouette_score is :  0.6314598904189876
For min sample value = 9 The average silhouette_score is :  0.6275777383195086
For min sample value = 10 The average silhouette_score is :  0.6243088264063699
For min sample value = 11 The average silhouette_score is :  0.6192288770135049
For min sample value = 12 The average silhouette_score is :  0.6345603012166445
For min sample value = 13 The average silhouette_score is :  0.6337546080736745
For min sample value = 14 The average silhouette_score is :  0.6304606820537793
For min sample value = 15 The average silhouette_score is :  0.6287638262850304
For min sample value = 16 The average silhouette_score is :  0.6284944013629604
For min sample value = 17 The average silhouette_score is :  0.6250433248061671
For min sample value = 18 The average silhouette_score is :  0.6156591612741592
For min sample value = 19 The average silhouette_score is :  0.6103836043473915
For min sample value = 20 The average silhouette_score is :  0.6077942134189246
For min sample value = 21 The average silhouette_score is :  0.6013389570633625
For min sample value = 22 The average silhouette_score is :  0.5861570217526623
For min sample value = 23 The average silhouette_score is :  0.5852061011592705
For min sample value = 24 The average silhouette_score is :  0.5795881861967718
For min sample value = 25 The average silhouette_score is :  0.561175267208149
```

Fig. 2: Grid Search for MinPts

Fig. 3 illustrates the workflow of the enhanced DBSCAN algorithm, termed SS-DBSCAN. In our study, we utilized two versions of datasets: Synthetic dataset and a Real-world dataset. Following the preprocessing, both datasets underwent a series of computational steps. We began by setting number of neighbors k to compute the Euclidean distances between any two elements in X. From this, we derived the average distance. Stratified sampling was then employed to ascertain each element's average distance from its nearest neighbors.

Subsequently, we carried out a polynomial fitting to construct a k-distance graph, which aided in pinpointing the eps value using the knee locator. We prefered k-distance method to determine the eps value in SS-DBSCAN because it offers robustness to noise, adaptability to local density variations, automatic selection, and flexibility to handle diverse datasets

effectively. It aligns with the core principles of DBSCAN, which is designed for density-based clustering in complex and non-uniform data distributions. We then executed a grid search technique with the eps value in hand to determine the optimal MinPts value. This process was done by holding the eps value constant and testing a range of values, starting from 3 up to n. These two parameters, eps and MinPts, were then input into the DBSCAN algorithm to identify the number of clusters. We used silhouette scores to gauge the fitting of MinPts. The score that rendered the highest value was selected. Importantly, in cases where the clustering outcome produced a silhouette score below 1, we systematically improved our results by adjusting the number of neighbors k. This change directly influenced the distances between any two elements X, which, in turn, subtly adjusted the eps and MinPts values.

4 Results

To assess the robustness of our methodology, we conducted experiments using two distinct types of datasets: synthetic datasets and real-world datasets, as elaborated in this section.

4.1 Experiments from the synthetic dataset

2d-20c-no0 Dataset: The 2d-20c-no0 dataset is a synthesized collection comprising 1,517 data points, each defined by attributes X and Y. By implementing our novel method; we derived an epsilon value of roughly 1.01432 and a MinPts of 13. This configuration produced a clustering silhouette score of 0.635, indicating a robust clustering arrangement Fig. 5. For benchmarking, the standard DBSCAN was deployed with eps set to 0.40150 and MinPts at 4, leading to a silhouette score of 0.418 Fig. 4. Thus, although both techniques showcased good performance, our method demonstrated superior clustering precision, resulting in 19 clusters that match the actual dataset labels and 58 noise points, compared to DBSCAN's 35 clusters with 173 noise points.

sizes1 Dataset: The sizes1 dataset comprises 1,000 instances, each defined by two features: X and Y. Utilizing our proposed method, we estimated an epsilon value of approximately 2.78358 and determined a MinPts value of 69. Applying these parameters to the SS-DBSCAN algorithm led to identifying 4 well-defined clusters, along with 18 noise points. These results align with the dataset's actual 4 class labels. Additionally, the clustering outcome achieved a silhouette score of 0.569 Fig. 7, which suggests a well-structured cluster arrangement. On the other hand, the DBSCAN algorithm, from the calculated eps=1.93697 and default MinPts value of 4. This approach yielded a lower silhouette score of 0.146 Fig. 6, with a single cluster and 3 noise points. Consequently, our method demonstrated a substantially better performance in clustering effectiveness.

elly_2d10c13s Dataset: The elly_2d10c13s dataset contains 2,796 instances with two features, X and Y. Using our method with SS-DBSCAN, we obtained an epsilon value of about 0.53009 and a MinPts value of 124. This resulted in a single cluster along with 102 noisy data points, although its performance fell short of expectations despite a respectable silhouette score of 0.450 Fig. 9. In contrast, standard DBSCAN with different parameters yielded a lower silhouette score of 0.249 Fig. 8. Overall, clustering was not effective for this dataset.

square4 Dataset: The square4 dataset consists of 1,000 instances, each characterized by two features, X and Y. Using our proposed method, we estimated an epsilon value of approximately 2.63267 and selected a MinPts value of 122. When these parameters were applied to the SS-DBSCAN algorithm, we identified 4 well-defined clusters and detected 86 noise points. These results correspond to the square4 dataset's actual labels. The clustering

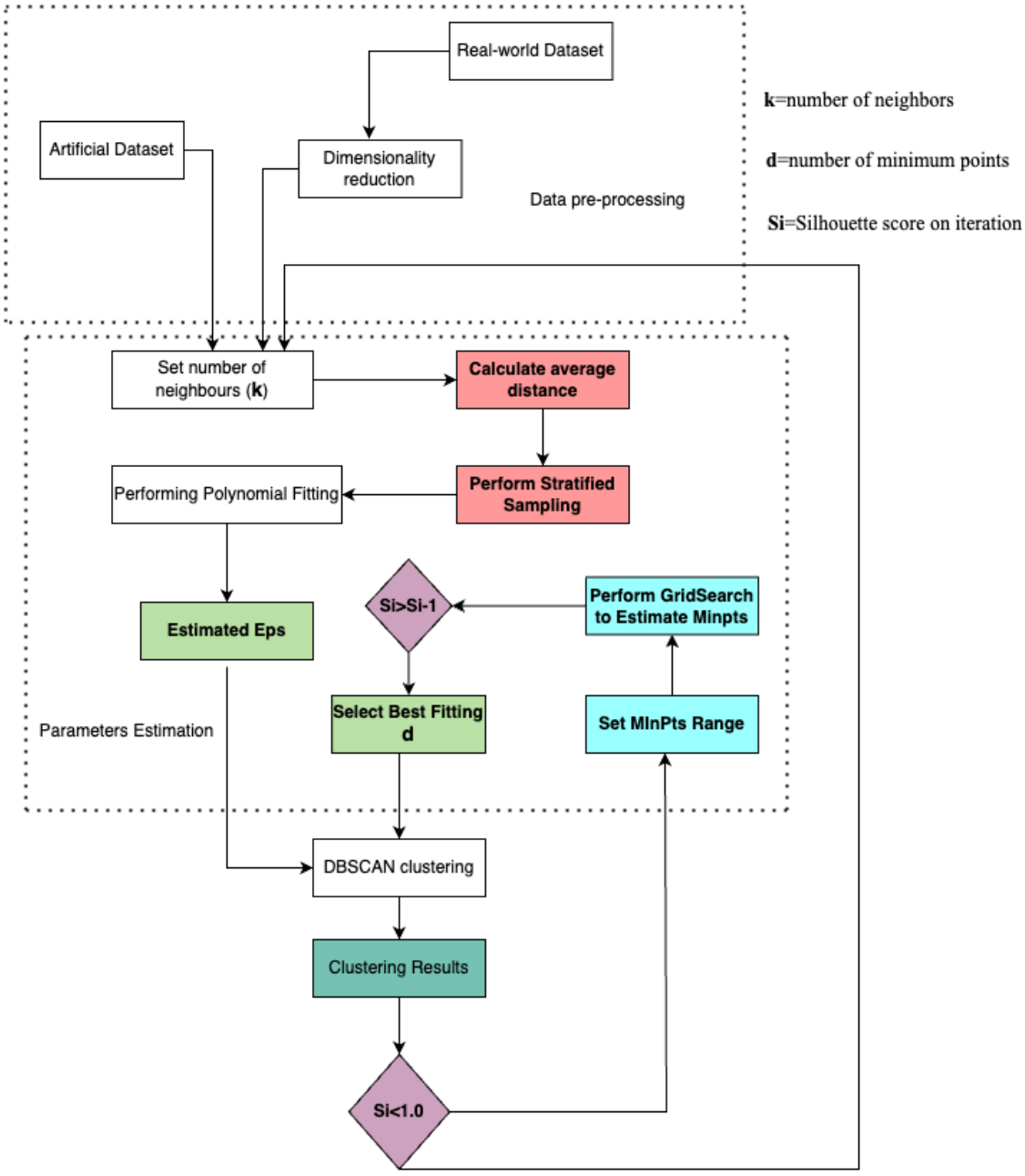

Fig. 3: SS-DBSCAN Flow Diagram

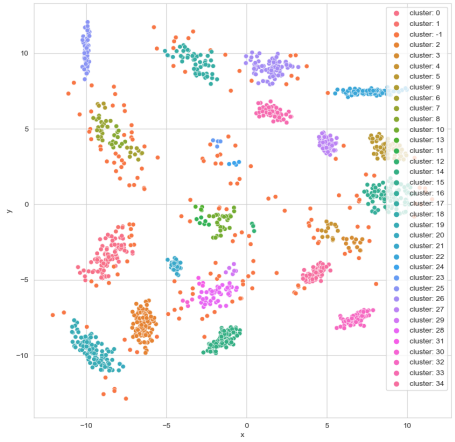

Fig. 4: DBSCAN=0.418

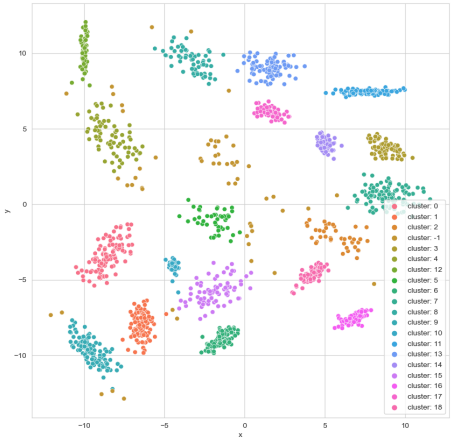

Fig. 5: SS-DBSCAN=0.635

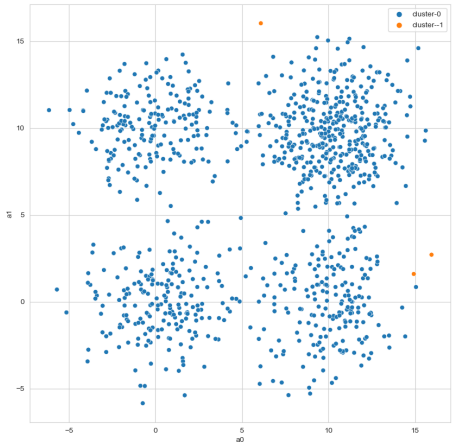

Fig. 6: DBSCAN= 0.146

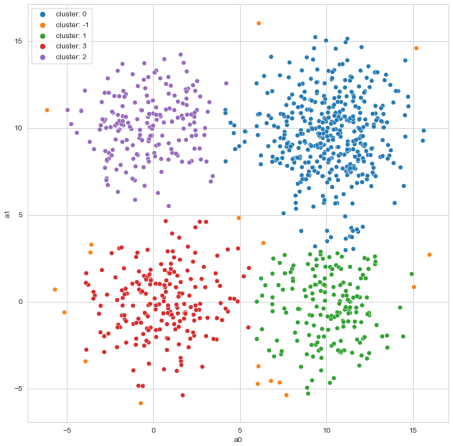

Fig. 7: SS-DBSCAN=0.569

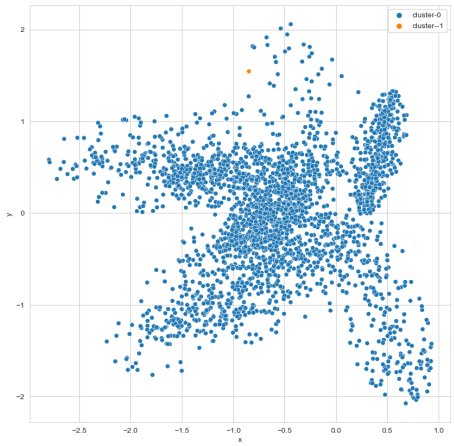

Fig. 8: DBSCAN= 0.249

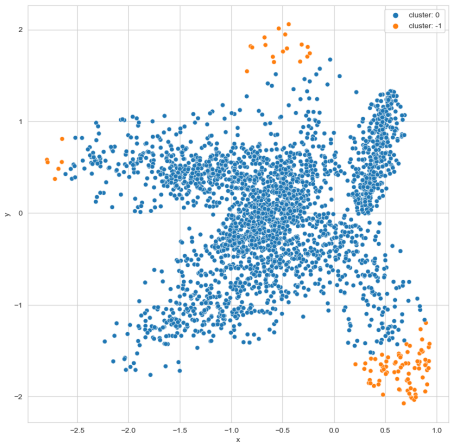

Fig. 9: SS-DBSCAN=0.450

outcome yielded a silhouette score 0.402, indicating a well-structured cluster arrangement Fig. 11. Compared with the DBSCAN algorithm, with parameters set at eps=1.57385 and MinPts=4. This approach resulted in a lower silhouette score of 0.295 and produced only one cluster and 6 noise points Fig. 10. Hence, our approach showcased exceptional performance in the context of clustering effectiveness.

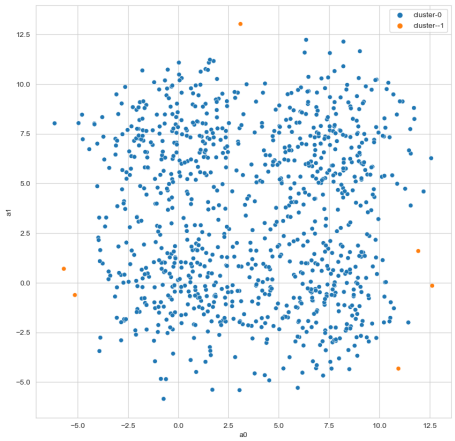

Fig. 10: DBSCAN=0.295

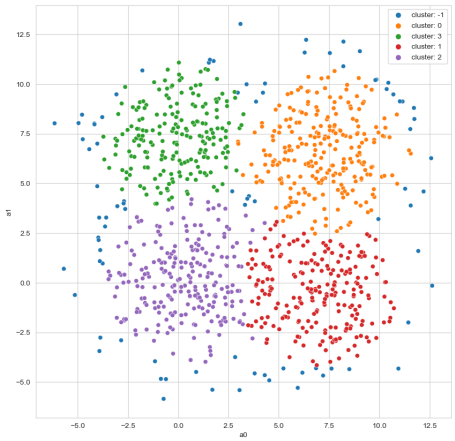

Fig. 11: SS-DBSCAN=0.402

st900 Dataset: The st900 dataset comprises 900 instances, each described by two features: X and Y. Utilizing our proposed method, we estimated an epsilon (eps) value of approximately 0.68156 and selected a MinPts value of 45. In this case, the SS-DBSCAN algorithm successfully identified 8 clusters and detected 124 noise points, while DBSCAN identified only one cluster. SS-DBSCAN achieved a silhouette score of 0.342, suggesting a reasonably well-structured cluster arrangement Fig. 13. We also implemented the DBSCAN algorithm using parameters eps=0.45121 and MinPts=4. This approach yielded a significantly lower silhouette score of 0.114 Fig. 12. Consequently, our proposed method demonstrated substantially superior performance in terms of clustering effectiveness.

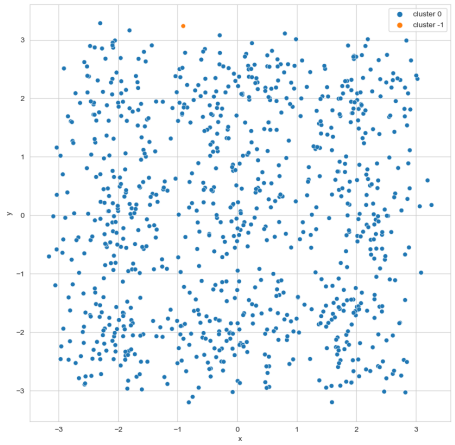

Fig. 12: DBSCAN=0.114

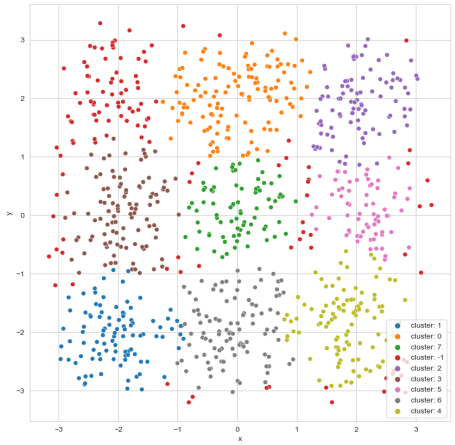

Fig. 13: SS-DBSCAN= 0.342

4.2 Experiments from real-world dataset

From our experimental analysis, real-world datasets displayed a consistent pattern, although the results were generally suboptimal except for the Iris dataset. This dataset exhibited superior performance across both the SS-DBSCAN and conventional DBSCAN algorithms. Notably, all these real-world datasets (Iris, Iono, Sonar, and Arrhythmia) have a significant number of dimensions, featuring 5, 35, 61, and 263 attributes, respectively. Our analysis revealed a notable trend: the SS-DBSCAN algorithm consistently outperformed the conventional DBSCAN algorithm across these diverse datasets. Nonetheless, it was noted that the excellence in cluster formation tended to diminish with the augmentation of dataset dimensionality, implying that although SS-DBSCAN exhibits enhanced performance accross all dataset compared to DBSCAN, its efficacy might encounter limitations when dealing with datasets of higher dimensions.

Iris dataset Fig. 14 and Fig. 15

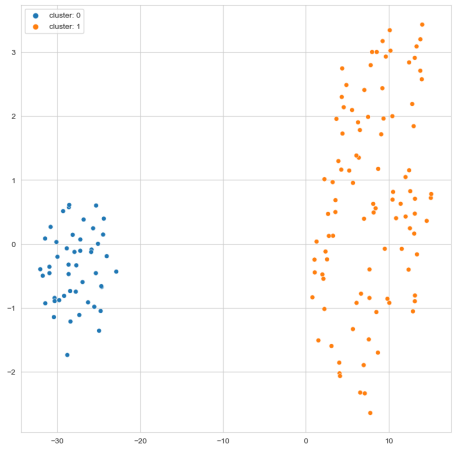

Fig. 14: DBSCAN=0.872

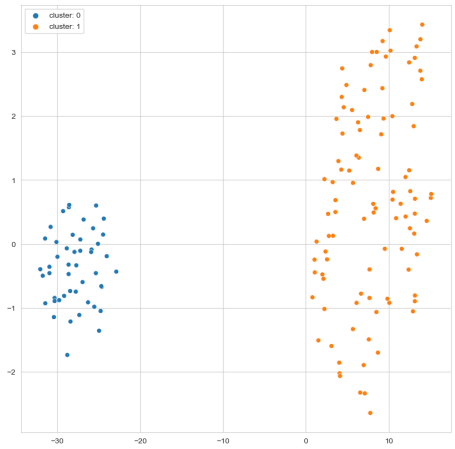

Fig. 15: SS-DBSCAN=0.872

Iono dataset Fig. 16 and Fig. 17

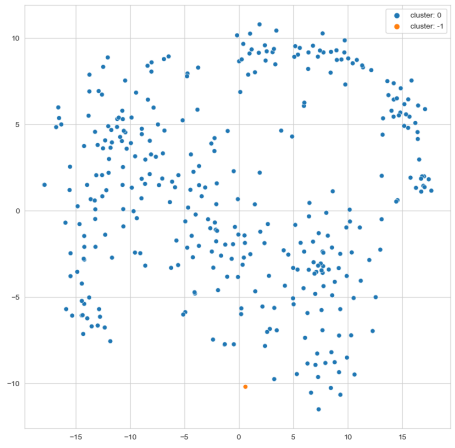

Fig. 16: DBSCAN=-0.183

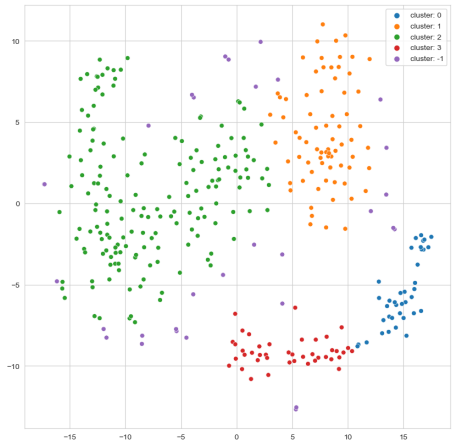

Fig. 17: SS-DBSCAN=0.385

Sonar dataset Fig. 18 and Fig. 19

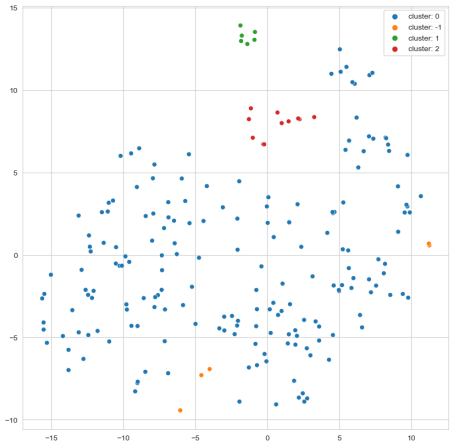

Fig. 18: DBSCAN=0.022

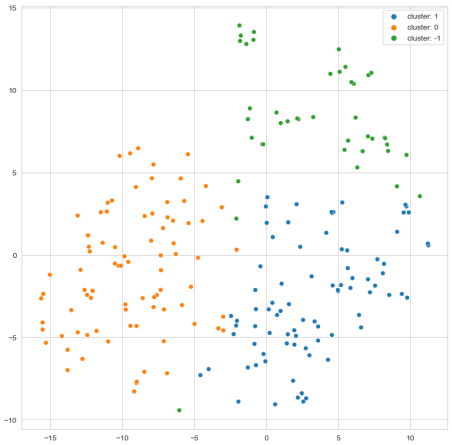

Fig. 19: SS-DBSCAN=0.440

Arrhythmia dataset Fig. 20 and Fig. 21

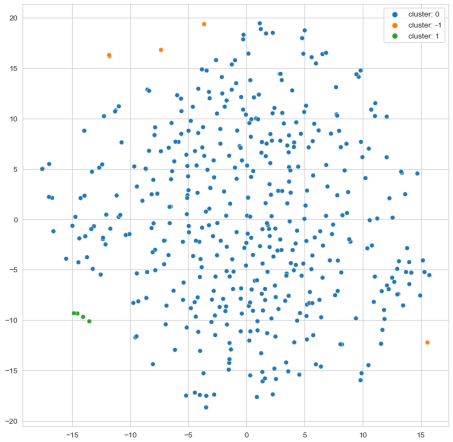

Fig. 20: DBSCAN=0.100

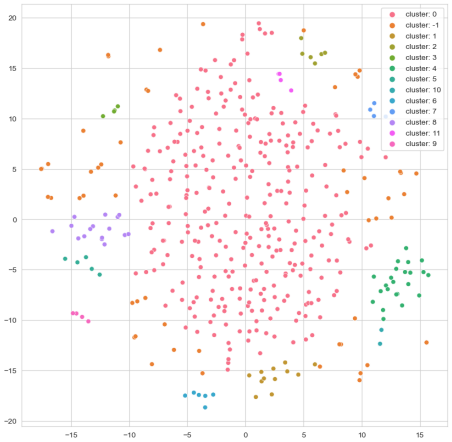

Fig. 21: SS-DBSCAN=0.177

4.3 SS-DBSCAN Validation

We employed several metrics in validating the robustness of our algorithm SS-DBSCAN Table 1, which are the Silhouette Score (SS), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), Homogeneity Score (HS), and Completeness Score (CS) to ensure a comprehensive assessment of our clustering results.

– a) Silhouette analysis is utilized to pinpoint the optimal cluster count for continuous scale data, like Euclidean distances, and it excels when clusters are clearly distinct. Rousseeuw [23]first presented the Silhouette as a tool to assess clustering performance. This technique considers both the likeness and differences or average closeness, between clusters. It's especially beneficial for clusters that are roughly spherical [24]. In our

study, we also leveraged silhouette analysis to corroborate the validity of our approach. There major three steps happens when validating using silhouette [23] [24].

$$a\left(i\right)=\frac{1}{\left|CI\right|-1}\sum_{J\in CI,i\neq j}d\left(i,j\right)$$

First, the centroid is taken into account. Let's label this centroid as i and the neighboring data points as j. We then determine the distances between i and j, denoted as $\mathrm{d}(i,j)$.
Next, we turn our attention to another cluster and measure the distances between these clusters using a specific formula

$$b\left(i\right)=\min\frac{1}{J\neq I\left|Cj\right|}\sum_{J\in CJ}d\left(i,j\right)$$

Lastly, The efficacy of the clustering model is gauged by:

$$b\left(i\right)>>a\left(i\right)$$

Assigning a silhouette value to a specific data point, denoted as i.

$$s\left(i\right)=\frac{b\left(i\right)-a\left(i\right)}{\max\left\{a\left(i\right),b\left(i\right)\right\}},\ if\ \ \left|CI\right|>1$$

If the silhouette values predominantly lie closer to 1 within a range from -1 to 1, it indicates a robust model.

– b) Normalized Mutual Information (NMI) serves as an additional metric to gauge the resemblance between two cluster patterns or a comparison of a clustering to its baseline truth. It captures the mutual details shared between two clustering outcomes, producing values that range from 0 (suggesting entirely distinct clusterings) to 1 (indicating compatible clusterings with possible permutations). NMI's framework draws upon information theory principles, factoring in mutual information and entropy, to establish this similarity.
Mutual Information (MI) can be portrayed as a derivation of Shannon's entropy values and their conditional counterparts, as referenced in [25].

$$((I\left(X;Y\right)=H\left(X\right)-H\left(X|\,Y\right)\ \ =H\left(Y\right)-H\left(Y|\,X\right)$$

The boundaries for mutual information are outlined in [26]:

$$0\leq I\left(X,Y\right)\leq\min\left(H\left(X\right),H\left(Y\right)\right)$$

$$NMI\left(X,Y\right)=\frac{I\left(X;Y\right)}{\min\left(H\left(X\right),H\left(Y\right)\right)}$$

where

$$\left(X;Y\right)$$

is the mutual information between clusters $X and$Y

$$\left(H\left(X\right)andH\left(Y\right)\right)$$

are the entropies of the clusters $X and$Y, respectively.

– c) The Rand Index (RI) [27] and its refined counterpart, the Adjusted Rand Index (ARI) [28], rank among the most recognized metrics for evaluating the congruence between distinct partitions. Originating from the realm of clustering analysis, these indices offer a quantifiable measure of how similar or different two data groupings are. Over the years, both RI and ARI have been embraced across diverse domains and have become the go-to metrics for cluster validation, as evidenced by numerous studies and applications in fields cited in references [29].

$$\mathrm{RI}\left(P^{(1)}, P^{(2)}\right) = \frac{a+d}{a+b+c+d} = \frac{\binom{n}{2} + \sum_{u=1}^{k_1}\sum_{v=1}^{k_2} n_{uv}^2 - \frac{1}{2}\left(\sum_{u=1}^{k_1} n_{u.}^2 + \sum_{v=1}^{k_2} n_{.v}^2\right)}{\binom{n}{2}}$$

where a quantifies the pairs of points that share a segment in both partitions, while b tallies the pairs residing in separate segments across both partitions. On the other hand, c and d keep track of pairs of points that align in one partition's segment but diverge in the other, considering both possible arrangements, as delineated in reference [30].

$$ARI\left(P^{(1)}, P^{(2)}\right) = \frac{RI - E(RI)}{\max(RI) - E(RI)}$$

– d) Homogeneity and completeness scores can be concisely defined as follows: A clustering outcome exhibits homogeneity when every one of its clusters contains data points belonging exclusively to a singular class [31]. Conversely, a clustering outcome demonstrates completeness when all data points associated with a specific class are consolidated within a single cluster [31]. Homogeneity score is calculated as:

$$h = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{else} \end{cases}$$

Where:

$$H(C \mid K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|} \frac{a_{ck}}{N} \log \frac{a_{ck}}{\sum_{c=1}^{|C|} a_{ck}}$$

$$H(C) = -\sum_{c=1}^{|C|} \frac{\sum_{k=1}^{|K|} a_{ck}}{n} \log \frac{\sum_{k=1}^{|K|} a_{ck}}{n}$$

Completeness score is calculated as:

$$c = \begin{cases} 1 & \text{if } H(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{else} \end{cases}$$

Where:

$$H(K \mid C) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|} \frac{a_{ck}}{N} \log \frac{a_{ck}}{\sum_{k=1}^{|K|} a_{ck}}$$

$$H(K) = -\sum_{k=1}^{|K|} \frac{\sum_{c=1}^{|C|} a_{ck}}{n} \log \frac{\sum_{c=1}^{|K|} a_{ck}}{n}$$

In this context, N signifies the data points within a dataset, which can be divided into two subsets: a collection of classes denoted as C = ci|i = 1, . . ., n, and a set of clusters designated as K = ki|1, . . ., m. The clustering algorithm generates a contingency table denoted as A, representing the clustering result. Within this table, A = aij, with each aij indicating the count of data points belonging to class ci and present in cluster kj [30]

Table 1: Table 1: DBSCAN Parameters Vs SS-DBSCAN Parameters

	DBSCAN					SS-DBSCAN				
Datasets	**SS**	**NMI**	**HS**	**ARI**	**CS**	**SS**	**NMI**	**HS**	**ARI**	**CS**
2d-20c-no0	0.418	0.883	0.905	0.774	0.862	0.635	0.971	0.973	0.963	0.969
elly-2d10c13s	0.249	0.001	0.001	0.000	0.389	0.450	0.064	0.035	0.011	0.481
sizes	0.146	0.003	0.002	0.000	0.109	0.569	0.870	0.885	0.905	0.856
square4	0.295	0.003	0.002	0.000	0.062	0.402	0.649	0.689	0.681	0.613
st900	0.114	0.002	0.001	0.000	0.263	0.342	0.758	0.756	0.709	0.760
Iris	0.872	0.734	0.776	0.568	1.000	0.872	0.734	0.776	0.568	1.000
Sonar	-0.022	0.125	0.103	-0.001	0.159	0.440	0.063	0.079	0.046	0.052
Arrhythmia	0.100	0.030	0.016	0.023	0.181	0.177	0.168	0.147	0.056	0.195
lono	-0.183	0.061	0.554	0.044	0.155	0.385	0.334	0.627	0.276	0.245

The experimental results obtained from nine diverse datasets unequivocally highlight the effectiveness of our proposed approach in precisely estimating the values of eps and MinPts for the SS-DBSCAN algorithm. Utilizing SS-DBSCAN for cluster identification consistently outperforms DBSCAN, as evidenced by superior scores across various validation metrics as shown in Fig. 22, Fig. 23, Fig. 24, Fig. 25 and Fig. 26.

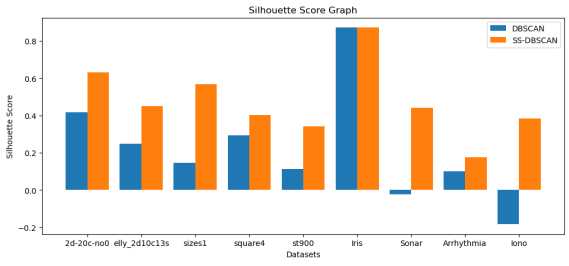

Fig. 22: Silhouette Score Graph

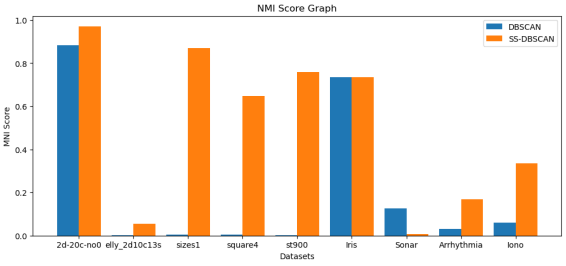

Fig. 23: NMI Score Graph

5 Discussion

This study introduced an enhanced methodology for estimating the eps and MinPts parameters in the DBSCAN clustering algorithm by leveraging stratified sampling combined with k-neighbors and Grid-search approaches. Drawing on experimental results from nine diverse datasets, we observed a marked improvement in the precision of eps and MinPt determination with our method when matched against established techniques.

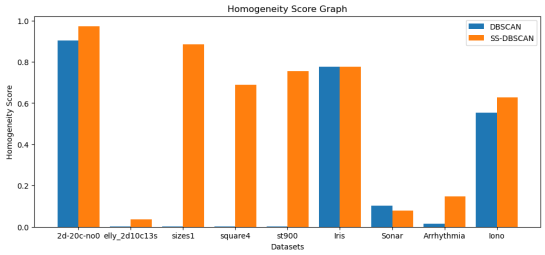

Fig. 24: Homogeneity Score Graph

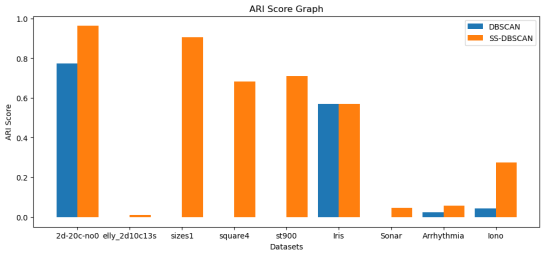

Fig. 25: ARI Score Graph

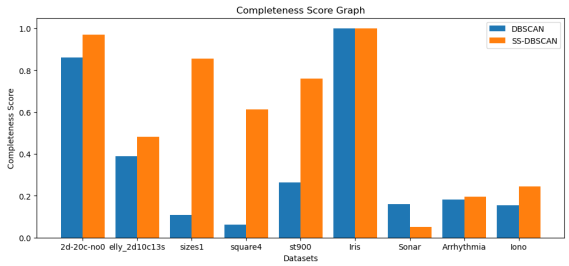

Fig. 26: Completeness Score Graph

5.1 Eps Estimation Through Stratified Sampling

A significant aspect of stratified sampling is its capacity to account for dataset structure and density variations. This consideration is instrumental in forming superior clusters. By constructing a k-nearest distance graph, we captured the intricate relationships between data points based on their relative proximities. This technique gave us the opportunity to explore deeper into the localized traits of each subgroup, furnishing a more comprehensive portrayal of relationships spanning the entire dataset. Furthermore, our approach underscores the essential contribution of each distinct stratum or subgroup within the dataset, ensuring that the insights extracted are comprehensive and do not overlook critical nuances specific to any particular subgroup.

5.2 MinPts Estimation Through Grid-Search

Traditionally, MinPts often uses rule-of-thumb values or formulae such as 4 or 2 times the dimensionality of the data. Our study argues against this practice, particularly in scenarios involving high-dimensional or non-uniformly dense datasets. Our grid-search methodology sets a flexible range for MinPts, maintains a constant, calculated eps value, and selects the best MinPts based on silhouette score metrics. Though our experiments commenced with a minimum value of 3, we note that this might not be universally applicable and caution that the range may need to be adjusted according to the specificities of the dataset in question, but the minimum number should be at least 3. The grid-search approach manifested its efficacy by delivering improved clustering results, not just in SS-DBSCAN but also showing promise for enhancing outcomes in conventional DBSCAN algorithms compared to default values.

5.3 Sensitivity of the SS-DBSCAN on Different Parameter Choice

The sensitivity analysis of SS-DBSCAN's parameters reveals that smaller sampling sizes tend to increase variance in density and distance estimations, potentially diminishing the

algorithm's robustness and leading to the detection of less quality clusters. Conversely, larger sampling sizes offer a more accurate data representation but may incur heightened computational demands. In terms of epsilon, reducing its value yields more clusters but can result in cluster fragmentation, while increasing it may merge clusters and compromise fine-grained structures. Striking a balance between sampling size and epsilon is essential, as smaller sizes may necessitate larger epsilon values, and vice versa. Achieving parameter robustness and generalization across diverse datasets is a pivotal objective, with evaluation metrics like silhouette scores aiding in quantifying the impact of parameter variations on clustering quality.

On the other hand, the selection of MinPts is primarily guided by the pursuit of the optimal silhouette score. However, situations may arise where multiple MinPts values yield identical silhouette scores and cluster counts. In such instances, it is prudent to opt for the lower MinPts value, as it tends to produce clusters with a reduced number of outliers, thus contributing to a more refined clustering outcome.

5.4 Robustness, Versatility, and Less Computational Overhead

A notable strength of our suggested method lies in its adaptability and resilience when applied to diverse datasets. Regardless of variations in scale, dimensionality, or density, our methodology consistently aligns with the unique attributes of each dataset, highlighting its widespread applicability and potential for extensive use. Another salient advantage of employing the stratified sampling approach is its efficacy in analysis. Focusing on targeted segments could expedite computational processes, presenting a significant reduction in computational overhead, especially pertinent in scenarios involving voluminous datasets. This streamlined approach not only improves the accuracy of the clusters but also augments the overall efficiency of the clustering process.

6 Conclusion

This research presented an innovative methodology to optimize parameter estimation within the DBSCAN clustering algorithm. By integrating stratified sampling and a k-neighbors approach, we achieved enhanced accuracy in eps parameter estimation, as evidenced by tests across nine diverse datasets. Our method outperforms traditional techniques and underscores the importance of a tailored approach to handle datasets of varying scales, dimensions, and densities.

Furthermore, our grid-search technique for MinPts estimation challenges conventional norms, highlighting the necessity for flexibility, especially in the face of high-dimensional or non-uniformly dense datasets. This flexible approach, grounded in silhouette score metrics, indicates a promising path forward for clustering algorithms, suggesting potential enhancements within our SS-DBSCAN variant and extending to traditional DBSCAN frameworks.

While our approach has demonstrated encouraging outcomes, forthcoming investigations could delve into the influence of alternative stratification methods, scrutinize scalability concerns, or assess the performance of the method on extensive datasets. Our study's findings accentuate the proposed methodology's robustness, versatility, and computational efficiency, emphasizing its broad applicability. As clustering remains a cornerstone in data analysis, the implications of our research are far-reaching, promising improved results across various domains.

References

1. X. X. Martin Ester, Hans-Peter Kriegel, Jiirg Sander, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in KDD-96 Proceedings, 1996, pp. 226–231.
2. M. M. A. Patwary, D. Palsetia, A. Agrawal, W. K. Liao, F. Manne, and A. Choudhary, "A new scalable parallel DBSCAN algorithm using the disjoint-set data structure," Int. Conf. High Perform. Comput. Networking, Storage Anal. SC, pp. 1–11, 2012, doi: 10.1109/SC.2012.9.
3. W. L. and A. C. Dianwei Han, Ankit Agrawal, A Fast DBSCAN Algorithm with Spark Implementation, vol. 44. Springer Singapore, 2018. doi: 10.1007/978-981-10-8476-8.
4. G. H. Shah, "An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional data sets," 3rd Nirma Univ. Int. Conf. Eng. NUiCONE 2012, pp. 1–6, 2012, doi: 10.1109/NUICONE.2012.6493211.
5. W. Lai, M. Zhou, F. Hu, K. Bian, and Q. Song, "A New DBSCAN Parameters Determination Method Based on Improved MVO," IEEE Access, vol. 7, pp. 104085–104095, 2019, doi: 10.1109/ACCESS.2019.2931334.
6. A. Karami and R. Johansson, "Choosing DBSCAN Parameters Automatically using Differential Evolution," Int. J. Comput. Appl., vol. 91, no. 7, pp. 1–11, 2014, doi: 10.5120/15890-5059.
7. J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications," Data Min. Knowl. Discov., vol. 2, no. 2, pp. 169–194, 1998, doi: 10.1023/A:1009745219419.
8. Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," Proc. - IEEE Int. Conf. Data Mining, ICDM, pp. 911–916, 2010, doi: 10.1109/ICDM.2010.35.
9. Y. Ren, X. Liu, and W. Liu, "DBCAMM: A novel density based clustering algorithm via using the Mahalanobis metric," Appl. Soft Comput. J., vol. 12, no. 5, pp. 1542–1554, 2012, doi: 10.1016/j.asoc.2011.12.015.
10. C. Tsai and C. Wu, "GF-DBSCAN: A New Efficient and Effective Data Clustering Technique for Large Databases," Proc. 9th WSEAS Int. Conf. Multimed. Syst. signal Process., no. January 2009, pp. 231–236, 2009, [Online]. Available: http://portal.acm.org/citation.cfm?id=1576697
11. X. Liu, Q. Yang, and L. He, "A novel DBSCAN with entropy and probability for mixed data," Cluster Comput., vol. 20, no. 2, pp. 1313–1323, 2017, doi: 10.1007/s10586-017-0818-3.
12. A. Starczewski, P. Goetzen, and M. J. Er, "A New Method for Automatic Determining of the DBSCAN Parameters," J. Artif. Intell. Soft Comput. Res., vol. 10, no. 3, pp. 209–221, 2020, doi: 10.2478/jaiscr-2020-0014.
13. Wiharto, A. K. Wicaksana, and D. E. Cahyani, "Modification of a Density-Based Spatial Clustering Algorithm for Applications with Noise for Data Reduction in Intrusion Detection Systems," Int. J. Fuzzy Log. Intell. Syst., vol. 21, no. 2, pp. 189–203, 2021, doi: 10.5391/IJFIS.2021.21.2.189.
14. J. C. Perafan-Lopez, V. L. Ferrer-Gregory, C. Nieto-Londoño, and J. Sierra-Pérez, "Performance Analysis and Architecture of a Clustering Hybrid Algorithm Called FA+GA-DBSCAN Using Artificial Datasets," Entropy, vol. 24, no. 7, pp. 1–19, 2022, doi: 10.3390/e24070875.
15. B. H. Shekar and G. Dagnew, "Grid search-based hyperparameter tuning and classification of microarray cancer data," 2019 2nd Int. Conf. Adv. Comput. Commun. Paradig. ICACCP 2019, pp. 1–8, 2019, doi: 10.1109/ICACCP.2019.8882943.
16. D. M. Belete and M. D. Huchaiah, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," Int. J. Comput. Appl., vol. 44, no. 9, pp. 875–886, 2022, doi: 10.1080/1206212X.2021.1974663.
17. E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," ACM Trans. Database Syst., vol. 42, no. 3, 2017, doi: 10.1145/3068335.
18. M. M. R. Khan, M. A. B. Siddique, R. B. Arif, and M. R. Oishe, "ADBSCAN: Adaptive density-based spatial clustering of applications with noise for identifying clusters with varying densities," 4th Int. Conf. Electr. Eng. Inf. Commun. Technol. iCEEiCT 2018, pp. 107–111, 2018, doi: 10.1109/CEEICT.2018.8628138.
19. D. Suchy and K. Siminski, "GrDBSCAN: A Granular Density-Based Clustering Algorithm," Int. J. Appl. Math. Comput. Sci., vol. 33, no. 2, pp. 297–312, 2023, doi: 10.34768/amcs-2023-0022.
20. M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," SIGMOD 2000 - Proc. 2000 ACM SIGMOD Int. Conf. Manag. Data, pp. 93–104, 2000, doi: 10.1145/342009.335388.
21. K. Giri, T. K. Biswas, and P. Sarkar, "ECR-DBSCAN: An improved DBSCAN based on computational geometry," Mach. Learn. with Appl., vol. 6, no. September, p. 100148, 2021, doi: 10.1016/j.mlwa.2021.100148.
22. A. B. Habib, "Elbow Method vs Silhouette Co-efficient in Determining the Number of Clusters Author: Adria Binte Habib," BRAC Univ., no. June, 2021, doi: 10.13140/RG.2.2.27982.79688.

23. P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," J. Comput. Appl. Math., vol. 20, no. C, pp. 53–65, 1987, doi: 10.1016/0377-0427(87)90125-7.
24. T. Thinsungnoen, N. Kaoungku, P. Durongdumronchai, K. Kerdprasop, and N. Kerdprasop, "The Clustering Validity with Silhouette and Sum of Squared Errors," pp. 44–51, 2015, doi: 10.12792/iciae2015.012.
25. A. Bingham, S. P. Arjunan, B. Jelfs, and D. K. Kumar, "Normalised mutual information of high-density surface electromyography during muscle fatigue," Entropy, vol. 19, no. 12, pp. 1–14, 2017, doi: 10.3390/e19120697.
26. S. Fehr and S. Berens, "On the conditional Rényi entropy," IEEE Trans. Inf. Theory, vol. 60, no. 11, pp. 6801–6810, 2014, doi: 10.1109/TIT.2014.2357799.
27. W. M. Rand, "Objective criteria for the evaluation of clustering methods," J. Am. Stat. Assoc., vol. 66, no. 336, pp. 846–850, 1971, doi: 10.1080/01621459.1971.10482356.
28. L. Hubert and P. Arabie, "Comparing partitions," J. Classif., vol. 2, no. 1, pp. 193–218, 1985, doi: 10.1007/BF01908075.
29. C. Carpineto and G. Romano, "Consensus clustering based on a new probabilistic rand index with application to subtopic retrieval," IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 12, pp. 2315–2326, 2012, doi: 10.1109/TPAMI.2012.80.
30. D. Stewart, A. Hampton, A. Zare, J. Dale, and J. Keller, "the Weakly-Labeled Rand Index," Int. Geosci. Remote Sens. Symp., vol. 2021-July, pp. 2313–2316, 2021, doi: 10.1109/IGARSS47720.2021.9553182.
31. A. Rosenberg and J. Hirschberg, "V-Measure: A conditional entropy-based external cluster evaluation measure," EMNLP-CoNLL 2007 - Proc. 2007 Jt. Conf. Empir. Methods Nat. Lang. Process. Comput. Nat. Lang. Learn., no. June, pp. 410–420, 2007.

Authors

Gloriana J. Monko received a Master of Information and Communication Science and Engineering from The Nelson Mandela African Institution of Science and Technology (NM-AIST), and a Bachelor's degree in Informatics from Sokoine University of Agriculture (SUA). She is currently pursuing her PhD in Engineering Science from Shibaura Institute of Technology. Her research interests include computer science, machine learning and natural language.

Prof. Masaomi Kimura received Ph.D. degree from The University of Tokyo. After his career of a system engineer in IBM, he started his career as a researcher at Shibaura Institute of Technology (SIT) in 2004. Now, he is a professor in the Department of Computer Science and Engineering in the Faculty of Engineering, and Department of Electrical, Electronics and Computer Engineering in the Graduate School of Science and Engineering, SIT. His research interests are in the areas of data science and data engineering, with a particular focus on data analysis as an application of artificial intelligence (machine learning), especially using deep learning.