Data Science 0 (2024) 1–0
IOS Press


# Enhanced SS-DBSCAN Clustering Algorithm for High-Dimensional Data

Gloriana Monko [a,*] and Masaomi Kimura [b]
[a] *Department of Functional Control Systems, Shibaura Institute of Technology, Toyosu, Koto, Tokyo, Japan*
*E-mail: nb22504@shibaura-it.ac.jp; ORCID: https://orcid.org/0000-0003-2414-5004*
[b] *School of Engineering, Shibaura Institute of Technology, Toyosu, Koto, Tokyo, Japan*
*E-mail: masaomi@shibaura-it.ac.jp; ORCID: https://orcid.org/0000-0003-3991-4259*

**Abstract.** This research introduces an enhanced SS-DBSCAN, a scalable and robust density-based clustering algorithm designed to tackle challenges in high-dimensional and complex data analysis. The algorithm integrates advanced parameter optimization techniques to improve clustering accuracy and interpretability. Key innovations include a Fast Grid Search (FGS) method for optimizing the search of optimal MinPts by keeping the $\epsilon$ parameter obtained constant. Notably, this study emphasizes the often-overlooked MinPts parameter, introducing a dynamic approach that initiates by calculating density metrics within a specified $\epsilon$ distance and adjusting the MinPts range based on the standard deviation of these metrics. This approach identifies optimal MinPts values based on the maximum allowed range. Comprehensive experiments on five real-world datasets demonstrate SS-DBSCAN's superior performance compared to DBSCAN, HDBSCAN, and OPTICS, evidenced by higher silhouette and Davies-Bouldin Index scores. The results highlight SS-DBSCAN's ability to capture intrinsic clustering structures accurately, providing deeper insights across various research domains. SS-DBSCAN's scalability and adaptability to diverse data densities make it a valuable tool for analyzing large, complex datasets.

Keywords: SS-DBSCAN Clustering, High-Dimensional, Fast Grid Search, Scalability, Adaptability

## 1. Introduction

Data mining is an interdisciplinary field that merges database technology, statistics, machine learning, and pattern recognition, benefiting from each of these areas [1]. While still not extensively adopted in many research domains, numerous studies have highlighted the potential of data mining in developing predictive models, evaluating risks, and assisting with decision-making [2]. Data mining utilizing large datasets can generate crucial and impactful insights that are vital for precise decision-making and risk evaluation [3]. Algorithms designed for data mining facilitate the achievement of these objectives.

The advent of large and complex datasets has ushered in a new era of data-driven insights across various domains. Among the myriad available datasets, those that encompass extensive and high-dimensional data like medical information stand out due to their comprehensive and detailed collection of information [4], [5]. The complexity, volume, and high dimensionality of these datasets pose significant challenges for clustering and data analysis, necessitating advanced methodologies for effective data preprocessing and clustering parameter optimization.

*Corresponding author. E-mail: nb22504@shibaura-it.ac.jp.

When managing datasets characterized by high density, arbitrary shapes, and irregular distribution, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is recommended as a robust algorithm specifically designed to address these complex scenarios [6]-[9]. However, DBSCAN has its limitations, particularly in the selection of its two primary parameters, which can affect its performance and accuracy [10]. Despite its effectiveness in handling complex datasets, DBSCAN faces challenges, particularly in the selection and tuning of its two key parameters: the minimum number of points required to form a dense region (MinPts) and the maximum distance between two points for one to be considered as in the neighborhood of the other ($\epsilon$) [11]. Among the two parameters, $\epsilon$ has been the subject of extensive research, whereas MinPts has often been overlooked, with its selection frequently relying on rule-of-thumb methods or manual estimation based on data size. However, both parameters play a crucial role in determining clustering outcomes. In particular, improper determination of MinPts can significantly affect clustering results, especially as the data size of the same dataset increases.

Other variants of DBSCAN, such as HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) and OPTICS (Ordering Points To Identify the Clustering Structure), have been developed to address some of these limitations. HDBSCAN extends DBSCAN by converting it into a hierarchical clustering algorithm that does not require the user to specify a fixed value for $\epsilon$, aiming to find clusters of varying densities. However, HDBSCAN still struggles with high-dimensional datasets, as the hierarchical approach can become computationally expensive and less effective in distinguishing between closely spaced clusters in such complex data [12], [13]. Similarly, OPTICS improves upon DBSCAN by ordering points to identify the clustering structure and handling clusters of varying densities more effectively. Nevertheless, OPTICS also faces challenges in high-dimensional spaces, where the complexity of data can lead to suboptimal clustering results and increased computational costs [14].

This paper elaborates on the enhancements introduced to the SS-DBSCAN algorithm from our previous work, focusing on its innovative approach to automatically select DBSCAN parameters [11], [12]. The methodology presented here is specifically tailored to navigate the complexities inherent in high-dimensional datasets, providing a more nuanced and effective clustering solution for the unique challenges posed by these extensive datasets. We introduce a more convenient and improved grid search method, named Fast Grid Search (FGS) for determining MinPts. By leveraging automatic parameter selection, SS-DBSCAN aims to improve the precision and applicability of clustering techniques, enhancing the potential for actionable insights in various research domains. Additionally, we demonstrate the pivotal role of Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) in preprocessing, alongside a modified approach to SS-DBSCAN parameter optimization, in enhancing clustering accuracy and interpretability within complex datasets. SS-DBSCAN is tested against other DBSCAN variant algorithms to demonstrate its robustness and resilience.

## 2. Related Works

Clustering algorithms, particularly Density-Based Spatial Clustering of Applications with Noise (DBSCAN), have been extensively studied for their capability to identify natural groupings in data without requiring a predefined number of clusters. The original DBSCAN algorithm, introduced by Ester et al. (1996), [9] demonstrated effectiveness in discovering clusters of arbitrary shapes and handling noise. Still, its performance heavily relies on the appropriate selection of two key parameters: $\epsilon$ and MinPts.

Numerous subsequent studies have attempted to address these challenges through various enhancements to the DBSCAN algorithm. Schubert et al. (2017) revisited DBSCAN and discussed that DB-

SCAN is still a practical and effective clustering algorithm, especially when applied with careful consideration of parameters and indexing strategies [10]. Selecting the $\epsilon$ parameter for DBSCAN in high-dimensional data is still challenging due to diminished contrast in distances [15]-[17]. This issue persists irrespective of the indexing method, making DBSCAN parameterization difficult in high-dimensional contexts. Algorithms like OPTICS and HDBSCAN eliminate the need for the $\epsilon$ parameter, making them more user-friendly. However, they also face challenges when dealing with high-dimensional data. [18]-[20].

Other modifications of the DBSCAN algorithm have been proposed to enhance its clustering performance. Liu et al. (2010) introduced DBSCAN-DLP, which uses a dynamic approach to select the $\epsilon$ value by calculating it for each data point based on the local density and mean distance, although this increases computational complexity [21]. Karami & Johansson (2014) developed BDE-DBSCAN, combining Binary Differential Evolution with DBSCAN to fine-tune its parameters [22], while Ren et al. (2012) created DBCAMM, which uses Mahalanobis distance and an innovative merging strategy for better image segmentation. [23] Lai et al. (2019) proposed an optimization technique using the MVO algorithm to iteratively refine DBSCAN parameters, [24] and Khan et al. (2018) introduced adaptive DBSCAN to automate parameter selection [25]. Despite these advancements, there is still a need for more adaptable and user-friendly methods for attaining better clusters with DBSCAN, which the current paper aims to address.

Other experiments conducted by Gan and Tao were performed on datasets, and their parameter settings were not well-suited for cluster analysis. Gan & Tao's (2015) choice of the $\epsilon$ ($\epsilon$) parameter was unusually large, set at a minimum of $\epsilon = 5{,}000$ for all their experiments [26]. Their results only demonstrated better performance under certain questionable settings. In contrast, more realistic parameter choices showed that SS-DBSCAN implementations by Monko & Kimura (2023) with an effective selection of both parameters (i.e. $\epsilon$ & MinPts) result in the best results [11].

In high-dimensional datasets, the complexity and volume of data present significant challenges for traditional clustering algorithms [27]-[29]. These datasets often contain intricate patterns that are not readily apparent, necessitating the use of advanced dimensionality reduction techniques like Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) [30]-[33]. In practice, applying PCA to reduce the dimensionality to a smaller number of components (e.g., 30-50) before running t-SNE is a common approach. This ensures that t-SNE works more efficiently and effectively, especially with very large or complex datasets [34], [35].

Various studies have also highlighted the potential of data mining and clustering in the medical domain. For instance, Zhang et al. (2016) explored big data mining in clinical medicine, emphasizing the utility of clustering techniques in identifying meaningful patterns in patient data [36]. Ngiam and Khor (2019) discussed the role of machine learning algorithms in healthcare delivery, underscoring the importance of robust clustering methods for clinical decision-making [2].

From the above-discussed related works, we realized a need to address the existing gaps by building upon SS-DBSCAN, a new variant of DBSCAN that we developed in our previous studies that incorporates stratified sampling for $\epsilon$ estimation and a novel grid search method for determining MinPts. We put more emphasis on the MinPts which most algorithms tend to use the rule of thumb 2 for 2 dimension or 2* D for high dimension where D is the dimension of the data. Abdulhameed et al. (2024) offers significant improvements over traditional DBSCAN (Semi Supervised-DBSCAN) by incorporating a pre-specified condition or constraint to better identify core points, the authors determined the MinPts parameter based on whether the dataset is noisy or not [37]. For noisy data, the MinPts is set to 2*D,

where D is the number of features and for noiseless data, the MinPts is set to D+1 [37]. These approaches still doesnt work well with complex real world data with high dimension. And upon increasing the data size it usually result into poorer clusters. In this work the MinPts determination is improved and offers better results than all other algorithms.The dual optimization of $\epsilon$ and MinPt sensures that SS-DBSCAN is finely tuned to the intrinsic clustering structures within the high-dimensional datasets, enhancing clustering accuracy and interpretability.

## 3. Contibution

This paper makes four main contributions to the field of data mining and clustering high-dimensional datasets:

(1) We enhanced the original DBSCAN to SS-DBSCAN to address the complexities of high-dimensional data through advanced parameter optimization techniques, ensuring precise and reliable clustering results.
(2) We developed a novel adaptive range method based on local density estimates. This method dynamically adjusts the range for MinPts to improve adaptability to varying data densities.
(3) We improved grid search technique that significantly reduces the computational time for finding optimal MinPts.
(4) With enhancement and improvement listed above, we realized scalable and adaptable DBSCAN, SS-DBSCAN, which is a valuable tool for analyzing large and complex datasets across various research domains.

## 4. Methodology

Our methodology outlines the strategies for achieving high-quality clustering using SS-DBSCAN. Fig. 1 illustrates the process through data pre-processing techniques and parameter selection, particularly emphasizing the MinPts parameter, which has historically been the most challenging to optimize in previous research.

### *4.1. Data Preprocessing*

Our preprocessing pipeline initiates with the application of Principal Component Analysis (PCA) to reduce the dimensionality of the datasets. These datasets comprise sequences with lengths ranging from a minimum of 50 to a maximum of 500, with an average length of 250. With PCA, we aim to retain the maximum variance with a minimized number of dimensions, simplifying the data while preserving the essential characteristics necessary for effective clustering.

Following PCA, we employed t-Distributed Stochastic Neighbor Embedding (t-SNE) to map the high-dimensional data into a two-dimensional space. In this context, applying PCA before t-SNE was essential for several reasons. Firstly, we used PCA to reduce the dimensionality of the data, making t-SNE computationally more efficient and faster. This also helped in reducing noise by focusing on the most significant features, thereby improving the input quality for t-SNE. Additionally, we used PCA to prevent t-SNE from overfitting to noise in high-dimensional data, leading to more robust and stable results. By preprocessing the data with PCA, t-SNE effectively uncovers and represents the underlying structure

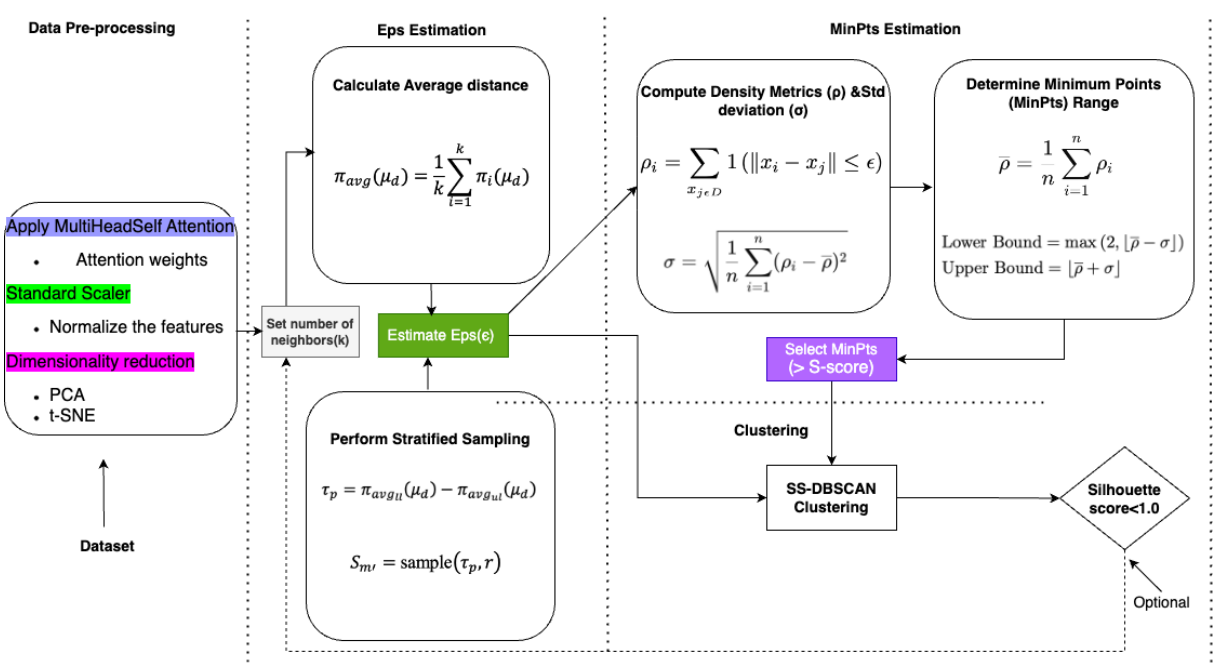



Fig. 1. Impoved MinPts for SS-DBSCAN Architecture

of the data, especially in large or complex datasets. This step is crucial for visualizing and clustering the data, as it helps to identify patterns that are not apparent in higher dimensions.

We used the all-mpnet-base-v2 model from Sentence-BERT to generate embeddings [38]-[40]. This pre-trained model, which has a maximum sequence length of 384 and dimensions of 768, has been trained over 1 billion training pairs [41]. By generating high-quality embeddings, we enhance the representational capacity of our data, facilitating more accurate and meaningful clustering.

### *4.2. SS-DBSCAN Parameter Selection*

For the clustering component, our algorithm, Stratified Sampling DBSCAN (SS-DBSCAN), employs a novel stratified sampling technique for estimating the $\epsilon$ parameter. This technique critically accommodates the unique density distributions found in the datasets, ensuring that $\epsilon$ is set to a value that accurately reflects the spatial distributions of the data points, thereby enhancing the natural clustering tendency.

#### *4.2.1. Fast Grid Search for MinPts*

To optimize the selection of the minimum points (MinPts), which dictate the core points in the DBSCAN algorithm, we implement a Fast Grid Search strategy. This approach tests a range of MinPts values to pinpoint the optimal number that maximizes cluster validity, as measured by silhouette scores. This metric assesses how similar an object is to its own cluster compared to other clusters [42]-[44]. In our previous work, we employed a grid search technique to determine the optimal value for MinPts. We manually established a range, starting from 3 and extending to a maximum value, iterating by 1 or in steps of $n$, while maintaining the $\epsilon$ value derived from the k-distance graph, which varied based on the data size and number of neighbors, $k$ [11]. This approach enabled partial automation in selecting MinPts;

however, we still had to manually define the range. This process not only increased execution time due to multiple iterations but also introduced the risk of overlooking critical values that could yield optimal silhouette scores, as the manual range setting might exclude such values.

Our novel method for selecting a single, optimal value for MinPts overcomes these limitations. It employs an adaptive range based on local density estimates. By calculating the density metrics $\rho_i$ for points within a specified $\epsilon$ distance and computing the standard deviation ($\sigma$)of these metrics, we dynamically adjust the range for MinPts by defining lower and upper bound. The lower bound is the average density minus the standard deviation, while the upper bound is the average density plus the standard deviation. This method significantly improves the adaptability of SS-DBSCAN to varying data densities and sizes. It provides a robust criterion for other analytical methods that require dynamic adjustments of sample sizes based on data density. Attaining the optimal value for MinPts involves a process described as follows:

- *Calculating Density Metrics*
  This function calculates the density metric for each point in a dataset by counting how many points lie within a certain distance $\epsilon$ of each point (1). For a dataset $D$ with points $x_i$, the density metric $\rho_i$ for point $x_i$ is given by:

$$\rho_i = \sum_{x_{j \epsilon D}} 1 \left( \|x_i - x_j\| \leqslant \epsilon \right), \tag{1}$$

  where,
  1 is the indicator function, which is 1 if the condition is true and 0 otherwise, and $\|x_i - x_j\|$ is the Euclidean distance between points $x_i$ and $x_j$.

$$\overline{\rho} = \frac{1}{n} \sum_{i=1}^{n} \rho_i, \tag{2}$$

  where, $\overline{\rho}$ is the mean of the density metrics, and $n$ is the total number of points (2)
- *Computing Standard Deviation*
  We then compute the standard deviation of the density metrics to understand the variability or spread of the density metrics across the dataset (3). If $\rho$ represents the vector of density metrics across all points, the standard deviation $\sigma$ is computed as:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\rho_i - \overline{\rho})^2}, \tag{3}$$

- *Computing the Range for Minimum Samples*
  Equation (4) computes a range for the MinPts parameter in SS-DBSCAN clustering based on the average density $\overline{\rho}$ and the computed standard deviation $\sigma$. The range is defined by a lower and upper bound, adjusted to ensure that the samples are at least 2.

$$\text{Lower Bound} = \max\left(2, \lfloor \overline{\rho} - \sigma \rfloor\right) \tag{4a}$$

$$\text{Upper Bound} = \lfloor \overline{\rho} + \sigma \rfloor \,. \tag{4b}$$

Here, $\lfloor . \rfloor$ denotes the floor function, which rounds down to the nearest integer. The function ensures that the lower bound is not less than 2, reflecting a minimum practical constraint for clustering.

- *Perform Fast Grid Search*
  After determining the optimal range for MinPts defined in Equation (4)a and (4)b, we employ a fast grid search technique to identify the best value by utilizing the silhouette score as a metric shown in Algorithm 1. This approach was enhanced by iterating through the identified range and, crucially, addressing the previous issue of unnecessarily printing all values within this range. To optimize the process, we introduced a stopping criterion: after identifying the current best MinPts, the iteration continues for five additional steps. If the silhouette score shows no improvement or consistently decreases during these iterations, the loop is terminated. This modification led to a significant reduction in computational time, decreasing execution from as long as 2,158 seconds to as short as 4 seconds in certain datasets, as demonstrated in Table 1. The execution time, however, increases proportionally with the data size, type and the number of neighbors ($k$), both of which directly influence the computation of the range.
  Through this dual optimization strategy ($\epsilon$ & MinPts), SS-DBSCAN is finely tuned to improve its sensitivity and adherence to the intrinsic clustering structures, thus promising more precise and reliable clustering outcomes.

**Algorithm 1** Find Optimal MinSamples for DBSCAN

**Input:** $features\_array\_tsne$: The t-SNE transformed features array
**Input:** $eps$: The epsilon value for SS-DBSCAN
**Input:** *start_min_samples*: The starting value of MinSamples
**Input:** *end_min_samples*: The ending value of MinSamples
**Output:** *best_min_samples*: The optimal MinSamples value
Initialize *best_silhouette_score* $\leftarrow -\infty$
Initialize *decrease_counter* $\leftarrow 0$
Initialize *last_silhouette_score* $\leftarrow -\infty$
**for** *i* $\leftarrow$ *start_min_samples* to *end_min_samples* **do**
    Perform DBSCAN clustering on $features\_array\_tsne$ with parameters:
        $eps \leftarrow eps$
        *min_samples* $\leftarrow$ *i*
    Calculate labels *labels* $\leftarrow$ *db.labels_*
    **if** number of unique clusters in *labels* $> 1$ **then**
        Calculate silhouette score *current_silhouette_score* $\leftarrow$ *silhouette_score*$(features\_array\_tsne, labels)$
        Print: "For min_samples value = " + str(*i*) + ", Total no of clusters = " + str(len(set(*labels*))) + ", Silhouette Score: " + str(*current_silhouette_score*)
        **if** *current_silhouette_score* > *best_silhouette_score* **then**
            Update *best_silhouette_score* $\leftarrow$ *current_silhouette_score*
            Update *best_min_samples* $\leftarrow$ *i*
            Reset *decrease_counter* $\leftarrow 0$
        **else**
            Increment *decrease_counter* $\leftarrow$ *decrease_counter* $+ 1$
            **if** *decrease_counter* $\geqslant 5$ **then**
                **break**
            **end if**
        **end if**
    **else**
        Print: "Insufficient clusters for min_samples value = " + str(*i*)
    **end if**
**end for**
Calculate *time_elapsed* $\leftarrow$ *time.time*$() -$ *since*
Print: "Time taken for training: :.0fm : :.0fs".format(*time_elapsed*//60, *time_elapsed*%60)
**return** *best_min_samples*

## 5. Experiment Setup

We designed a comprehensive experimentation process to rigorously evaluate the effectiveness of various clustering algorithms across multiple datasets, including MIMIC III, Emotion-Sentiment, Coronavirus-Tweets, Cancer-Doc, and Sonar. To demonstrate the scalability of SS-DBSCAN, we firstly, conducted experiments with varying data sizes of same dataset to assess its consistency and robustness,

addressing a common limitation of many clustering algorithms, which often struggle with performance degradation as data size increases. Secondly, we conducted experiments using various algorithms across multiple datasets.

The performance of each clustering algorithm was evaluated using the silhouette score, which measures the quality of clustering [42]-[44]. A higher silhouette score indicates better-defined and more distinct clusters, thereby validating the effectiveness of the clustering technique. We also utilized Davies-Bouldin Index (DBI), a metric used to evaluate the quality of clustering algorithms based on the concepts of cluster compactness and separation [45]. Lower DBI values (close to 0) indicate good clustering while higher DBI values (much greater than 1) indicate poorer clustering.

Table 1
Comparison of Grid Search and Fast Grid Search at Various Data Sizes

| **Data size** | **Search Range** | **Grid Search** | | **Fast Grid Search** | |
|---|---|---|---|---|---|
| | | **MinPts** | **Time(secs)** | **MinPts** | **Time(secs)** |
| 10000 | 525-1509 | 550 | 920 | 550 | 6 |
| 12000 | 1210-3150 | 1217 | 2158 | 1217 | 4 |
| 10000 | 1225-3506 | 1279 | 1980 | 1279 | 6 |
| 7000 | 755-1825 | 810 | 600 | 810 | 31 |
| 6000 | 95-557 | 292 | 480 | 292 | 2 |
| 5000 | 77-317 | 102 | 60 | 102 | 3 |
| 5000 | 345-1225 | 466 | 242 | 446 | 2 |
| 4000 | 50-184 | 64 | 4 | 64 | 2 |

### *5.1. Clustering Algorithms Applied in Different Data Sizes*

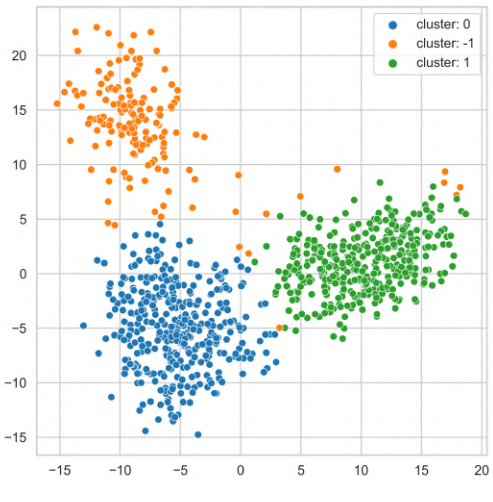
To evaluate the effectiveness of our preprocessing and clustering methodologies, we performed comparative analyses of several algorithms across varying data sizes within the MIMIC III dataset. The primary objective of this experiment was to demonstrate the robustness of SS-DBSCAN, particularly in managing complex datasets, and to assess its performance consistency as data size increases, an area where other algorithms often exhibit limitations as seen in Table 2. The MIMIC dataset used in this study primarily consists of two distinct clusters: Adverse Drug Reaction (ADR) and Non-ADR cases. Among the algorithms tested, only SS-DBSCAN consistently identified the correct number of clusters, regardless of increasing data size. In contrast, the other algorithms produced varying numbers of clusters with inconsistent results as the dataset expanded. These findings demonstrate that the enhanced SS-DBSCAN algorithm delivers superior clustering accuracy and robustness compared to the other algorithms evaluated in this experiment. Clustering results are also visualized in Fig. 2, Fig. 3, Fig. 4 and Fig. 5

#### *5.1.1. Clustering Results with SS-DBSCAN*

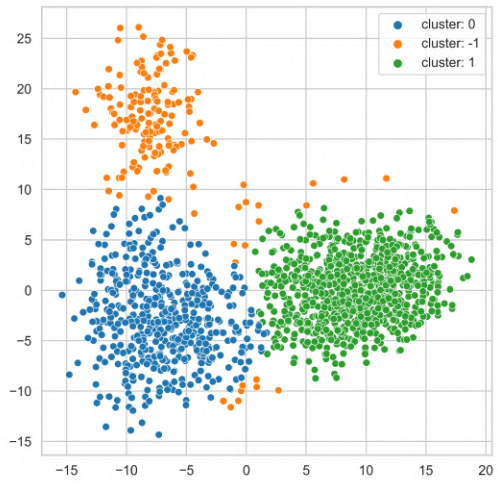
SS-DBSCAN employs stratified sampling for precise estimation of the $\epsilon$ parameter and utilizes a Fast Grid Search method for optimizing MinPts, allowing for the dynamic adjustment of DBSCAN's parameters to better align with the inherent structure of the data. The resulting parameter values vary accordingly based on the characteristics of the dataset. Figure 2 presents the clustering results generated by SS-DBSCAN.

Table 2
Parameter Values and Cluster Results of Different Algorithms at Various Data Sizes of MIMIC III

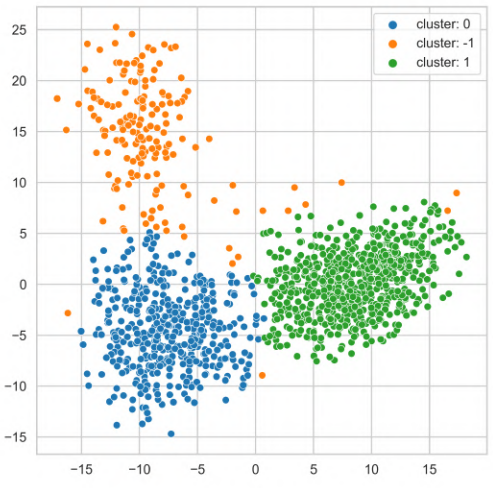
| DataSize | SS-DBSCAN | | | DBSCAN | | | HDBSCAN | | OPTICS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | eps | MinPt | Clusters | eps | MinPt | Clusters | ClusterSize | Clusters | xi | MinPt | Clusters |
| 1000 | 4.1189 | 93 | 2 | 1.7438 | 4 | 3 | 30 | 3 | 0.001 | 9 | 5 |
| 2000 | 3.3084 | 75 | 2 | 1.3950 | 4 | 10 | 20 | 3 | 0.001 | 7 | 5 |
| 3000 | 3.1629 | 59 | 2 | 1.0884 | 4 | 6 | 14 | 4 | 0.001 | 7 | 4 |
| 4000 | 3.2177 | 64 | 2 | 0.9509 | 4 | 16 | 15 | 4 | 0.001 | 8 | 4 |
| 5000 | 2.1996 | 102 | 2 | 0.8634 | 4 | 41 | 20 | 4 | 0.001 | 7 | 4 |



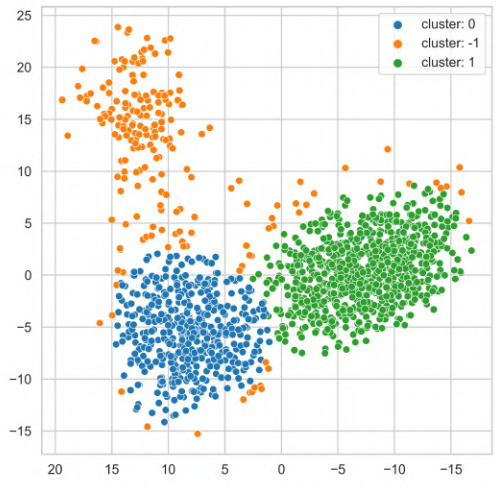

(a) Data size = 1000




(b) Data size = 3000




(c) Data size = 4000

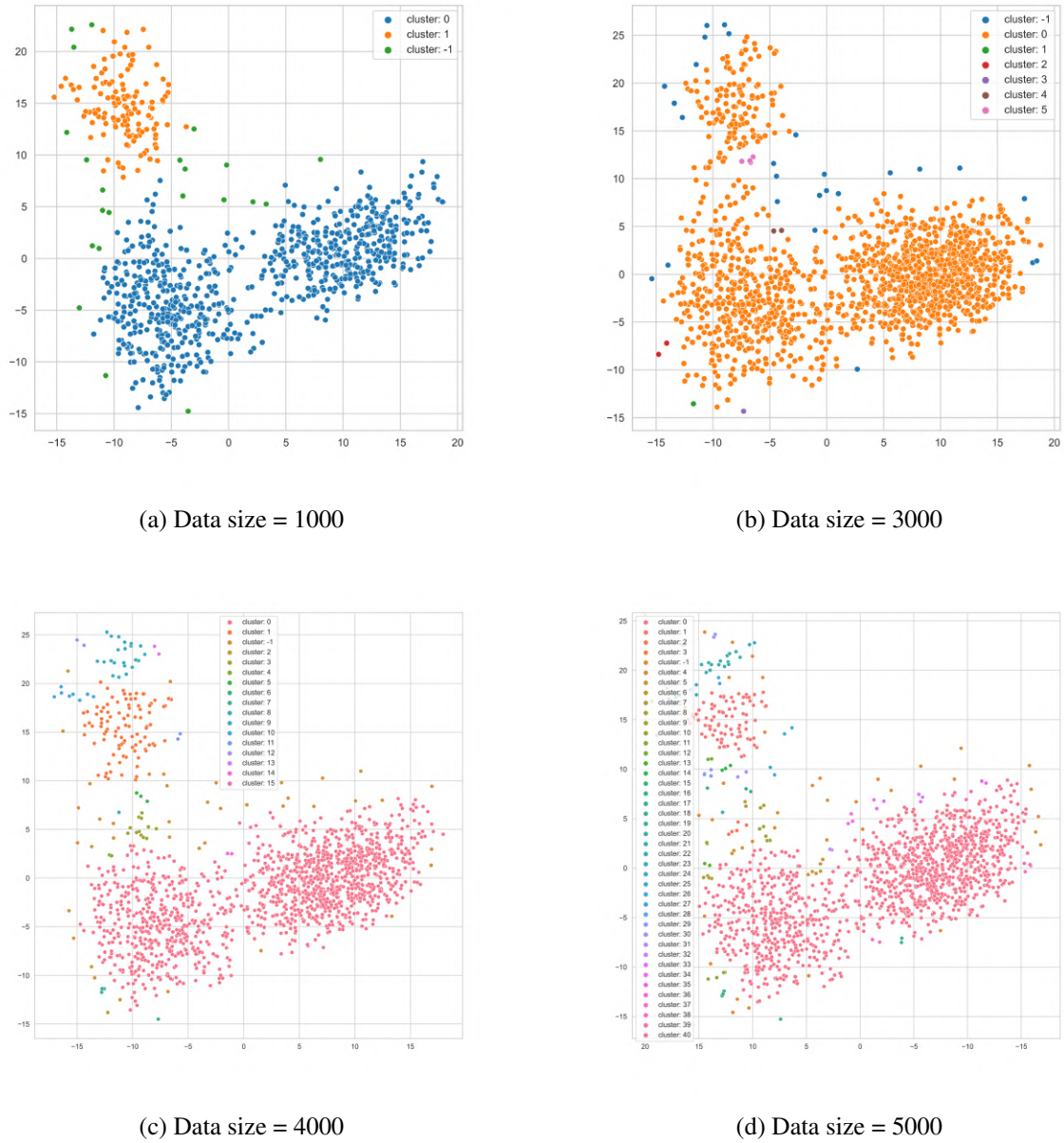



(d) Data size = 5000

Fig. 2. Comparison of SS-DBSCAN results for different data sizes.

### 5.1.2. Clustering Results with DBSCAN

DBSCAN utilizes manually selected parameters based on established practices, specifically setting MinPts to 4 and $\epsilon$ to the knee value. The $\epsilon$ parameter is determined by identifying the knee point, which corresponds to the location of a significant bend in the curve. The MinPts parameter is chosen according to the rule that suggests MinPts should be set to 2 times the dimensionality of the data (2*dim). Fig. 3 illustrates the clusters obtained.




(a) Data size = 1000

(b) Data size = 3000

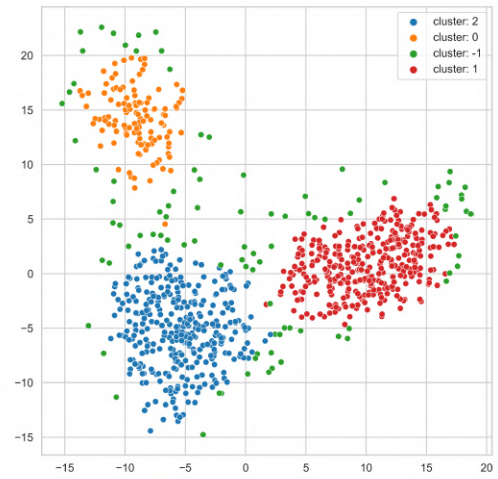
(c) Data size = 4000

(d) Data size = 5000

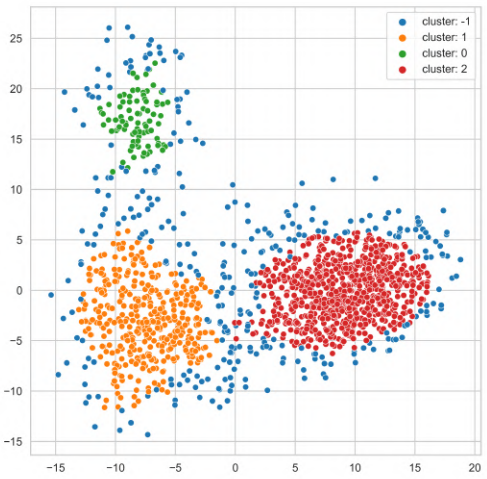
Fig. 3. Comparison of DBSCAN results for different data sizes.

### 5.1.3. *Clustering Results with HDBSCAN*

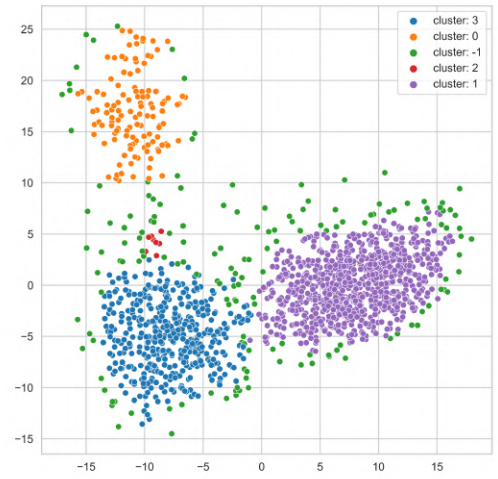
Another preferred algorithm is hierarchical DBSCAN (HDBSCAN), which can handle varying density clusters without specifying the epsilon or a global density threshold. Fig. 4 presents the cluster results for HDBSCAN.



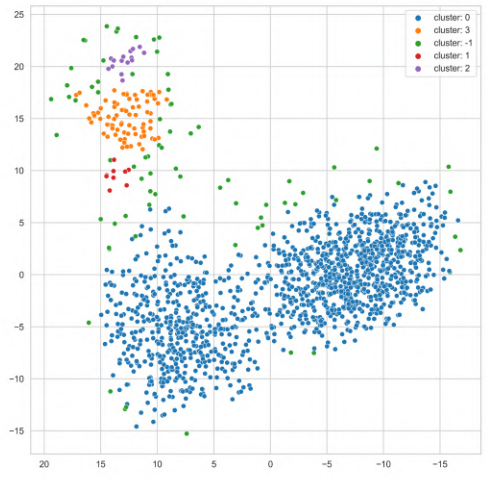

(a) Data size = 1000




(b) Data size = 3000




(c) Data size = 4000

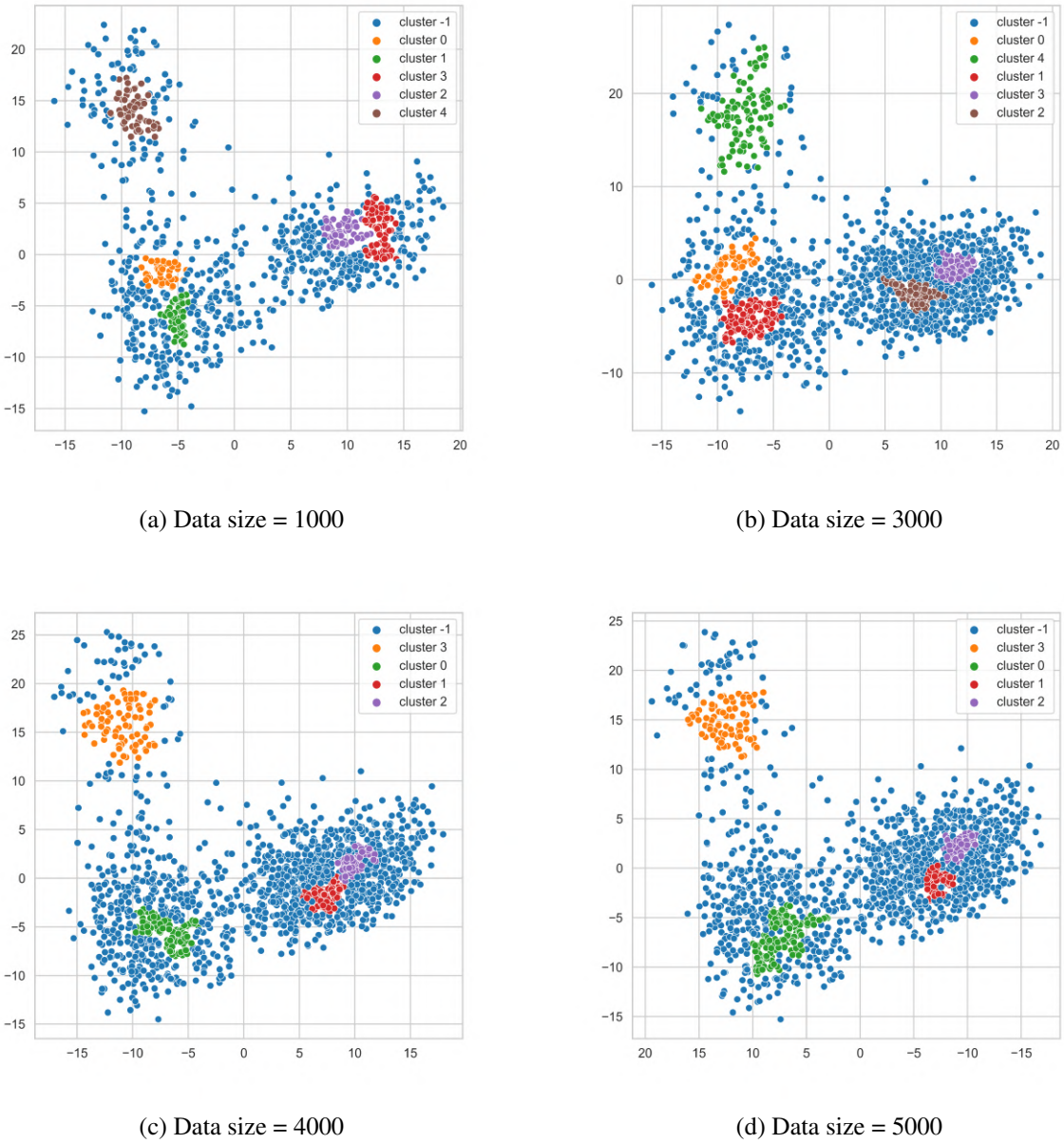



(d) Data size = 5000

Fig. 4. Comparison of HDBSCAN results for different data sizes.

### 5.1.4. Clustering Results with OPTICS

OPTICS (Ordering Points To Identify the Clustering Structure) is an extension of the DBSCAN algorithm designed to identify clusters in data with varying densities. Unlike DBSCAN, which relies on fixed parameters, OPTICS produces an ordered list of points based on their reachability distances, allowing it to reveal the clustering structure at multiple density levels. Fig. 5 shows the cluster results for OPTICS algorithm.




(a) Data size = 1000

(b) Data size = 3000

(c) Data size = 4000

(d) Data size = 5000

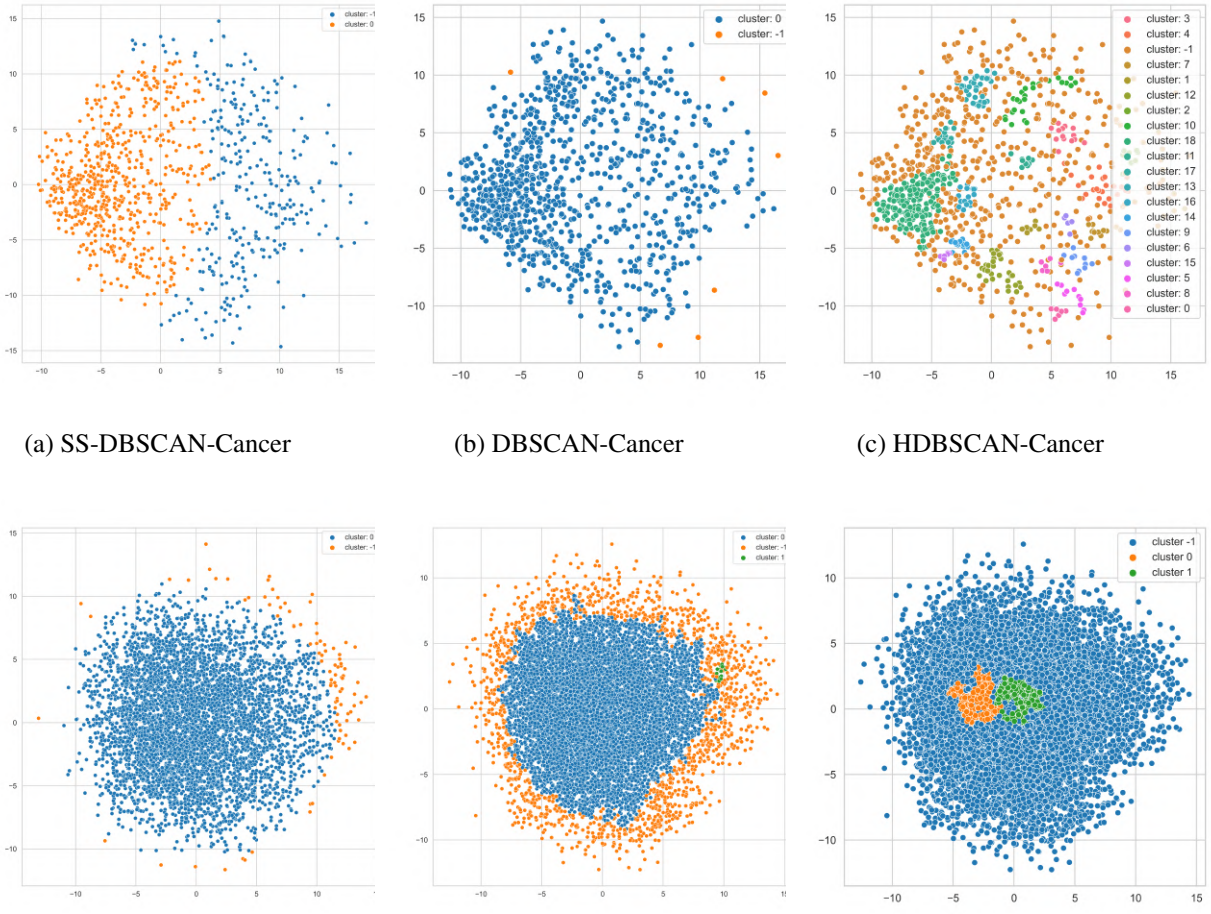
Fig. 5. Comparison of OPTICS results for different data sizes.

### 5.2. Results for Clustering Algorithms Applied in Different Datasets

In addition to experimenting with a single dataset of various sizes, we also conducted experiments across multiple datasets using all four algorithms. The datasets included Emotion-Sentiment, Coronavirus-Tweets, Cancer-Doc, and Sonar. Our comprehensive evaluation demonstrated that SS-DBSCAN consistently outperformed the other algorithms across all datasets and data sizes by the use of parameter values indicated in Table 3. The results of these experiments are presented in Fig. 8, providing a quantitative comparison of clustering performance. Additionally, the clustering outcomes are visualized better in Fig. 6, which illustrates the superior clustering quality achieved by SS-DBSCAN. These visualizations highlight the algorithm's ability to effectively manage varying data densities and complex structures, reinforcing its robustness and applicability across different types of high-dimensional data.

Table 3
Parameter Values and Cluster Results of Different Algorithms on Various Datasets

| **Dataset** | **SS-DBSCAN** | | | **DBSCAN** | | | **HDBSCAN** | | **OPTICS** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **eps** | **MinPt** | **Clusters** | **eps** | **MinPt** | **Clusters** | **Min Cluster Size** | **Clusters** | **xi** | **MinPt** | **Clusters** |
| Emotion Sentiment | 3.7833 | 466 | 1 | 1.0365 | 4 | 1 | 15 | 2 | 0.001 | 7 | 2 |
| Corona Tweets | 3.2270 | 194 | 2 | 1.1366 | 4 | 3 | 20 | 2 | 0.001 | 7 | 1 |
| CancerDoc | 3.5805 | 82 | 2 | 1.8789 | 4 | 1 | 15 | 19 | 0.001 | 7 | 7 |
| MIMIC III | 3.2177 | 64 | 2 | 0.9509 | 4 | 41 | 15 | 4 | 0.001 | 8 | 4 |
| Sonar | 5.0213 | 51 | 2 | 1.5947 | 4 | 1 | 15 | - | 0.001 | 7 | 7 |




(a) SS-DBSCAN-Cancer (b) DBSCAN-Cancer (c) HDBSCAN-Cancer

(d) SS-DBSCAN-Emotions (e) HDBSCAN-Emotions (f) OPTICS-Emotions

Fig. 6. Comparison of different algorithms across datasets.

### 5.3. *Comparative Analysis*

Each clustering technique is applied to the preprocessed data, and its performances are compared across algorithms with different data sizes and datasets. The clustering effectiveness is analyzed in the context of the data's size, complexity, and high dimensionality, taking into account the nuances and variability inherent in these datasets.

Our results in Fig. 7 and Fig. 8 demonstrate the effectiveness of SS-DBSCAN in achieving more reliable and meaningful clustering outcomes than other DBSCAN variants. More results are described in Table 4 and Table 5. The stratified sampling approach for $\epsilon$ estimation and the FGS for MinPts significantly enhance the robustness and resilience of the clustering process. This highlights the scalability of SS-DBSCAN and its adaptability to varying data densities and sizes and high dimensional data making it a valuable tool for complex data and decision-making.

Table 4
Silhouette Scores and Davies-Bouldin Index (DBI) of Different Algorithms at Various Data Sizes

| Data Size | SS-DBSCAN | | DBSCAN | | HDBSCAN | | OPTICS | |
|---|---|---|---|---|---|---|---|---|
| | Silhouette | DBI | Silhouette | DBI | Silhouette | DBI | Silhouette | DBI |
| 1000 | 0.64 | 0.39 | 0.24 | 1.41 | 0.54 | 1.51 | 0.02 | 1.37 |
| 2000 | 0.62 | 0.48 | -0.22 | 1.36 | 0.40 | 1.59 | -0.11 | 1.25 |
| 3000 | 0.61 | 0.44 | -0.40 | 1.45 | 0.38 | 1.50 | -0.16 | 1.23 |
| 4000 | 0.61 | 0.44 | -0.46 | 1.54 | 0.34 | 1.59 | -0.22 | 1.22 |
| 5000 | 0.61 | 0.45 | -0.04 | 1.52 | 0.09 | 1.57 | -0.35 | 1.25 |

Table 5
Silhouette Scores and Davies-Bouldin Index (DBI) of Different Algorithms on Various Datasets

| Dataset | SS-DBSCAN | | DBSCAN | | HDBSCAN | | OPTICS | |
|---|---|---|---|---|---|---|---|---|
| | Silhouette | DBI | Silhouette | DBI | Silhouette | DBI | Silhouette | DBI |
| EmotionsSentiments | 0.55 | 0.79 | 0.40 | 0.79 | 0.18 | 0.79 | -0.23 | 0.79 |
| CoronavirusTweets | 0.43 | 0.83 | 0.35 | 0.83 | -0.11 | 0.83 | -0.16 | 0.83 |
| CancerDoc | 0.45 | 0.71 | -0.40 | 0.71 | 0.09 | 0.71 | -0.21 | 0.71 |
| MIMIC III | 0.64 | 0.39 | -0.45 | 0.39 | 0.342 | 0.39 | -0.35 | 0.39 |
| Sonar | 0.56 | 0.40 | 0.35 | 0.40 | - | 0.40 | -0.06 | 0.40 |

## 6. Result Interpretation

Our implementation of SS-DBSCAN significantly enhances the clustering process by allowing us to precisely select the optimal values for $\epsilon$ (the maximum radius of the neighborhood) and MinPts (the minimum number of points in a neighborhood to form a cluster). This method ensures that we consistently achieve reliable clustering outcomes, distinctly improving upon the approach used by other density-based clustering algorithms.

In the standard DBSCAN framework, the MinPts parameter is typically determined using a heuristic based on the dataset's dimensionality, often set at twice the number of dimensions. In our study, after reducing the data's dimensionality from 768 to 2, we applied a MinPts value of 4 following this rule of thumb. However, this approach is somewhat arbitrary and may fail to accurately reflect the true density distribution in more complex datasets. Consequently, this led to suboptimal clustering results as shown in our experiments.

On the other hand, HDBSCAN, another variation of DBSCAN, adjusts its sensitivity based on several parameters such as min_cluster_size, min_samples, and alpha. The performance of HDBSCAN hinges significantly on the appropriate selection of min_cluster_size. Ineffective choices for this parameter can lead to poor clustering results, whereas optimal parameter tuning can considerably enhance the clustering quality. However, as the data size increases, the performance of HDBSCAN reduces, often returning meaningless clusters, as seen in Fig. 4.

OPTICS (Ordering Points To Identify the Clustering Structure) was also included in our experiment to explore its potential advantages over traditional density-based methods like DBSCAN and HDBSCAN. OPTICS attempts to uncover the clustering structure of data by ordering points based on their density-reachability. However, in our experiments, OPTICS underperformed in all datasets, as illustrated in Figure 5 and Figure 8. The algorithm's sensitivity to initial parameter settings (min_samples,

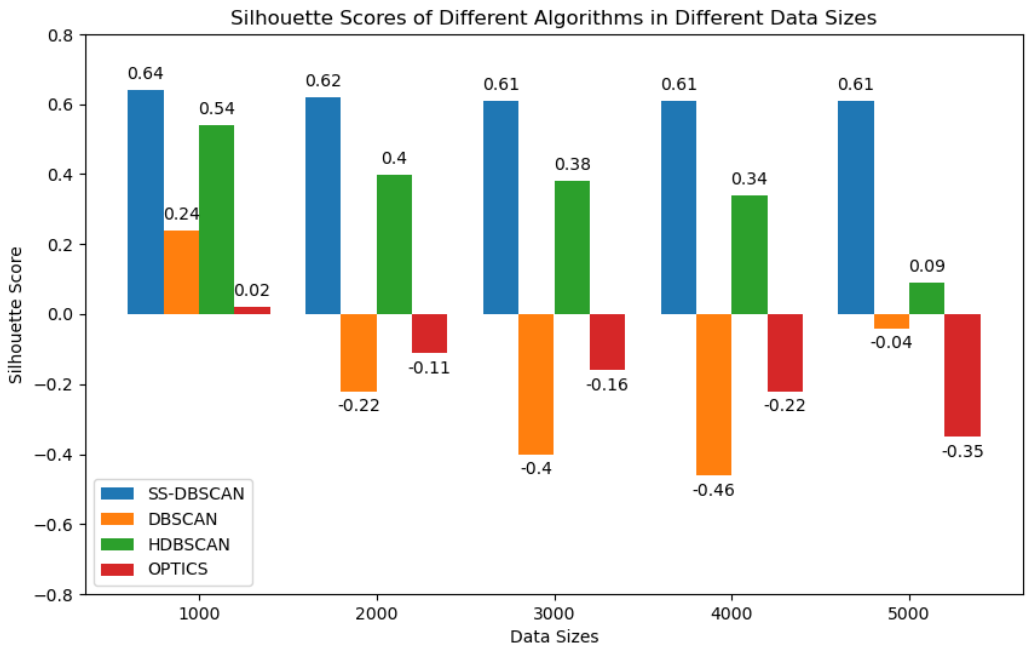



Fig. 7. Algorithms' Performance Compared in Different Data Sizes

xi, min_cluster_size, metric), coupled with its computational complexity, resulted in very poor clusters. OPTICS struggled to adapt to the intricate density variations in the data, ultimately producing less meaningful clustering outcomes.

Therefore, SS-DBSCAN distinguishes itself from other algorithms by incorporating stratified sampling to determine the best values for $\epsilon$ and FGS to determine MinPts without arbitrary estimations. This approach allows SS-DBSCAN to adapt effectively across different sizes and complexities of datasets, including complex and noisy datasets. SS-DBSCAN delivers consistent results and underscores our algorithm's robustness, making it highly effective for diverse applications.

## 7. Discussion

In this paper, we enhanced the SS-DBSCAN and evaluate its performance in different data sizes and on different datasets. Our findings underscore the adaptability and robustness of SS-DBSCAN, especially in handling large and complex data. The unique parameter optimization approach of SS-DBSCAN enhances its efficacy in identifying meaningful clusters vital for data mining and decision-making. Below, we discuss several aspects of SS-DBSCAN's application and the implications of our results:

### *7.1. Noise Sensitivity*

SS-DBSCAN performs better in managing noise in all datasets used for the experiment than traditional DBSCAN, HDBSCAN, and OPTICS. SS-DBSCAN ensures clearer and more relevant clusters by

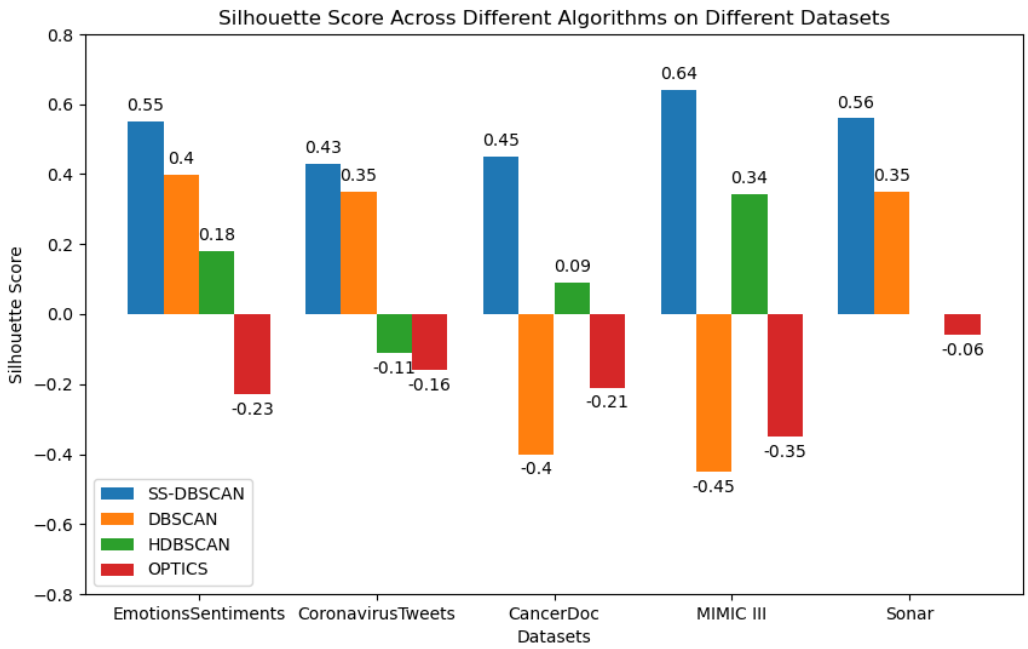



Fig. 8. Algorithms' Performance Compared in Different Datasets

effectively identifying and excluding noise-related data points. This capability particularly benefits large and complex datasets where outlier data can significantly skew results.

### *7.2. Scalability*

The scalability of SS-DBSCAN was rigorously evaluated using the MIMIC III dataset, a large and complex real-world dataset. The results demonstrate the algorithm's efficiency in handling extensive data volumes while maintaining high-quality clustering performance. This establishes SS-DBSCAN as a highly suitable solution for large-scale datasets where computational efficiency and time constraints are critical factors. Furthermore, the algorithm's flexibility in determining both the $\epsilon$ and MinPts parameters consistently yields more accurate and reliable results, regardless of dataset size. This adaptability underscores SS-DBSCAN's robustness across varying data densities, further enhancing its applicability in diverse research and real-world scenarios.

### *7.3. Parameter Adaptivity and Robusteness*

Our methodology dynamically adjusts $\epsilon$ and MinPts based on the dataset's intrinsic characteristics. This adaptivity allows SS-DBSCAN to respond flexibly to variations in data density and distribution, ensuring optimal clustering across different datasets. We also explored how variations in $\epsilon$ and MinPts in different datasets affect the stability of the clusters. Our results show that SS-DBSCAN maintains consistent clustering quality even with minor parameter adjustments, highlighting its reliability for clustering applications where precision is paramount.

### 7.4. Cluster Validation

We employed a silhouette statistical measure and Davies-Bouldin Index (DBI) to validate the clusters generated by SS-DBSCAN. Both silhouette and DBI scores confirm the distinctiveness and relevance of the clusters. In comparison with other algorithms used in our experiment, SS-DBSCAN stands out for its robustness and precision. Unlike methods that require extensive parameter tuning and may not form clusters effectively, SS-DBSCAN adapts its parameters automatically, offering more reliable clustering even in complex datasets.

## 8. Conclusion

We proposed enhanced SS-DBSCAN, which is a potent tool for clustering in various datasets, especially in complex domains. It is particularly suited to the complexities of large, noisy, and unspherical datasets. Its ability to adapt parameters dynamically, efficient handling of large data volumes, and robustness to noise make it an invaluable asset for clustering tasks and research.

By adeptly integrating PCA and t-SNE with the SS-DBSCAN parameter selection strategy, we have significantly enhanced the accuracy and interpretability of our clustering results. These improvements not only bolster the practicality of clustering analyses in complex datasets but also provide deeper, actionable insights into data. Such insights hold the potential to influence decision-making. Future research should explore the applicability of SS-DBSCAN to various data types, such as image datasets.

## References


[1] J. Iavindrasana, G. Cohen, A. Depeursinge, H. Müller, R. Meyer, and A. Geissbuhler, "Clinical data mining: a review.," *Yearb. Med. Inform.*, pp. 121–133, 2009, doi: 10.1055/s-0038-1638651.

[2] K. Y. Ngiam and I. W. Khor, "Big data and machine learning algorithms for health-care delivery," *Lancet Oncol.*, vol. 20, no. 5, pp. e262–e273, 2019, doi: 10.1016/S1470-2045(19)30149-4.

[3] W. T. Wu *et al.*, "Data mining in clinical big data: the frequently used databases, steps, and methodological models," *Mil. Med. Res.*, vol. 8, no. 1, pp. 1–12, 2021, doi: 10.1186/s40779-021-00338-z.

[4] A. Arya and R. Abhishek Arya B.E., "Exploratory Data Analysis of Intensive Care Unit Patients using MIMIC-III Database," 2019. [Online]. Available: https://www.minsal.cl/wp-content/uploads/2019/01/2019.01.23_PLAN-NACIONAL-DE-CANCER_web.pdf

[5] M. Mollura, G. Mantoan, S. Romano, L. W. Lehman, R. G. Mark, and R. Barbieri, "The role of waveform monitoring in Sepsis identification within the first hour of Intensive Care Unit stay," *2020 11th Conf. Eur. Study Gr. Cardiovasc. Oscil. Comput. Model. Physiol. New Challenges Oppor. ESGCO 2020*, pp. 1–9, 2020, doi: 10.1109/ESGCO49734.2020.9158013.

[6] 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.

[7] A. Ram, S. Jalal, A. S. Jalal, and M. Kumar, "A Density Based Algorithm for Discovering Density Varied Clusters in Large Spatial Databases," *Int. J. Comput. Appl.*, vol. 3, no. 6, pp. 1–4, 2010, doi: 10.5120/739-1038.

[8] 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.

[9] 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.

[10] 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.

[11] G. J. Monko and M. Kimura, "Optimized DBSCAN Parameter Selection: Stratified Sampling for Epsilon and Gridsearch for Minimum Samples," pp. 43–61, 2023, doi: 10.5121/csit.2023.132004.

[12] G. J. Monko and M. Kimura, “SS-DBSCAN: Epsilon Estimation with Stratified Sampling for Density-Based Spatial Clustering of Applications with Noise,” *Proc. - 2023 Int. Conf. Autom. Control Electron. Eng. CACEE 2023*, pp. 72–76, 2023, doi: 10.1109/CACEE61121.2023.00023.
[13] S. Fotopoulou, “A review of unsupervised learning in astronomy,” *A*stronomy and Computing, vol. 48, May 2024, doi: 10.1016/j.ascom.2024.100851.
[14] Y. F. Wang, Y. Jiong, G. P. Su, and Y. R. Qian, "A new outlier detection method based on OPTICS," *S*ustainable Cities and Society, vol. 45, pp. 197–212, 2019, doi: 10.1016/j.scs.2018.11.031.
[15] A. Hui and B. J. Gao, “When is Nearest Neighbor Meaningful: Sequential Data,” *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 3103–3106, 2021, doi: 10.1145/3459637.3482219.
[16] D. Hutchison and J. C. Mitchell, *Scientific and Statistical Database Management*.
[17] A. Zimek, E. Schubert, and H.-P. Kriegel, “A survey on unsupervised outlier detection in high-dimensional numerical data,” *Stat. Anal. Data Min.*, vol. 5, no. 5, pp. 363–387, 2012, [Online]. Available: http://dx.doi.org/10.1002/sam.11161.
[18] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander, “OPTICS: Ordering Points to Identify the Clustering Structure,” *SIGMOD Rec. (ACM Spec. Interes. Gr. Manag. Data)*, vol. 28, no. 2, pp. 49–60, 1999, doi: 10.1145/304181.304187.
[19] Z. Deng, Y. Hu, M. Zhu, X. Huang, and B. Du, “A scalable and fast OPTICS for clustering trajectory big data,” *Cluster Comput.*, vol. 18, no. 2, pp. 549–562, 2015, doi: 10.1007/s10586-014-0413-9.
[20] H. K. Kanagala and V. V. Jaya Rama Krishnaiah, “A comparative study of K-Means, DBSCAN and OPTICS,” *2016 Int. Conf. Comput. Commun. Informatics, ICCCI 2016*, pp. 1–6, 2016, doi: 10.1109/ICCCI.2016.7479923.
[21] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, “Understanding of internal clustering validation measures,” *Proc. - IEEE Int. Conf. Data Mining, ICDM*, no. Proc.-IEEE Int. Conf. Data Mining, ICDM, pp. 911–916, 2010, doi: 10.1109/ICDM.2010.35.
[22] 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.
[23] 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.
[24] 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.
[25] 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.
[26] J. Gan and Y. Tao, “DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation,” in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ACM, 2015, pp. 519–530. [Online]. Available: https://doi.org/10.1145/2723372.2737792.
[27] M. Paoletti *et al.*, “Explorative data analysis techniques and unsupervised clustering methods to support clinical assessment of Chronic Obstructive Pulmonary Disease (COPD) phenotypes,” *J. Biomed. Inform.*, vol. 42, no. 6, pp. 1013–1021, 2009, doi: 10.1016/j.jbi.2009.05.008.
[28] M. Saeed, C. Lieu, G. Raber, and R. G. Mark, “MIMIC II: A massive temporal ICU patient database to support research in intelligent patient monitoring,” *Comput. Cardiol.*, vol. 29, pp. 641–644, 2002, doi: 10.1109/cic.2002.1166854.
[29] S. Wang, M. B. A. McDermott, G. Chauhan, M. Ghassemi, M. C. Hughes, and T. Naumann, “MIMIC-Extract,” *ACM CHIL 2020 - Proc. 2020 ACM Conf. Heal. Inference, Learn.*, pp. 222–235, 2020, doi: 10.1145/3368555.3384469.
[30] H. Abdi and L. J. Williams, “Principal component analysis. wiley interdisciplinary reviews: computational statistics,” *Wiley Interdisplinary Rev. Comput. Stat.*, pp. 1–47, 2010.
[31] B. Melit Devassy and S. George, “Dimensionality reduction and visualisation of hyperspectral ink data using t-SNE,” *Forensic Sci. Int.*, vol. 311, p. 110194, 2020, doi: 10.1016/j.forsciint.2020.110194.
[32] A. Platzer, “Visualization of SNPs with t-SNE,” *PLoS One*, vol. 8, no. 2, 2013, doi: 10.1371/journal.pone.0056883.
[33] M. Smetana, L. Salles de Salles, I. Sukharev, and L. Khazanovich, “Highway Construction Safety Analysis Using Large Language Models,” *Appl. Sci.*, vol. 14, no. 4, 2024, doi: 10.3390/app14041352.
[34] J. Pareek and J. Jacob, “Data Compression and Visualization Using PCA and T-SNE,” Advances in Information Communication Technology and Computing, pp. 327–337, 2020.
[35] R. Shah and S. Silwal, “Using Dimensionality Reduction to Optimize t-SNE,” 2019. [Online]. Available: http://arxiv.org/abs/1912.01098.
[36] Y. Zhang, S. L. Guo, L. N. Han, and T. L. Li, “Application and exploration of big data mining in clinical medicine,” *Chin. Med. J. (Engl).*, vol. 129, no. 6, pp. 731–738, 2016, doi: 10.4103/0366-6999.178019.
[37] T. Z. Abdulhameed, S. A. Yousif, V. W. Samawi and H. I. Al-Shaikhli, "SS-DBSCAN: Semi-Supervised Density-Based Spatial Clustering of Applications with Noise for Meaningful Clustering in Diverse Density Data," in IEEE Access, doi: 10.1109/ACCESS.2024.3457587

[38] R. Korea and A. Zahran, "UNLPSat TextGraphs-16 Natural Language Premise Selection task: Unsupervised Natural Language Premise Selection in mathematical text using sentence-MPNet Premise Selection task: Unsupervised Natural Language Premise Selection in mathematical text using s," 2022.

[39] Y. He, Z. Yuan, J. Chen, and I. Horrocks, "Language Models as Hierarchy Encoders," 2024, doi: 10.5281/zenodo.10511042.

[40] S. M. Jayanthi, V. Embar, and K. Raghunathan, "Evaluating Pretrained Transformer Models for Entity Linking in Task-Oriented Dialog," 2021, [Online]. Available: http://arxiv.org/abs/2112.08327.

[41] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," *EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf.*, pp. 3982–3992, 2019, doi: 10.18653/v1/d19-1410.

[42] 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.

[43] 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.

[44] 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.

[45] Y. A. Wijaya, D. A. Kurniady, E. Setyanto, W. S. Tarihoran, D. Rusmana, and R. Rahim, "Davies bouldin index algorithm for optimizing clustering case studies mapping school facilities," TEM J., vol. 10, no. 3, pp. 1099-1103, 2021