universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

A Unified Perception of Density for Clustering

verfasst von | submitted by

Ekaterina Kuznetsova

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von | Supervisor:	Univ.-Prof. Dipl.-Inform.Univ. Dr. Claudia Plant
Mitbetreut von | Co-Supervisor:	Dr. Anna Beer BSc MSc

Abstract

Density-based clustering algorithms like DBSCAN are highly effective but sensitive to parameter selection, particularly the neighborhood radius (ε) and the minimum number of neighboring points to form a cluster (minPts). This research introduces an approach to automate parameter tuning by integrating persistent homology, a technique from topological data analysis. Persistent homology analyzes topological features, such as connected components and loops, across multiple spatial scales, improving clustering accuracy and robustness.

The study explores the relationship between density-based clustering and topological structures, using synthetic datasets with varying dimensionality, density, and noise levels. The results of the experiments demonstrate how insights from persistent homology can help to identify optimal parameter values by analyzing clustering outcomes and the corresponding ε-graphs across diverse settings. Furthermore, the research validates the robustness of this approach under different initial conditions and in the presence of noise.

The proposed technique improves the parameter selection process, allowing DBSCAN and related algorithms to perform effectively on different datasets. It combines topological insights with clustering techniques to provide a foundation for robust, scalable, and automated approaches to complex data analysis.

Kurzfassung

Dichtebasierte Clustering-Algorithmen wie DBSCAN sind hoch effektiv, aber sensitiv gegenüber der Wahl der Parameter, insbesondere des Nachbarschaftsradius (ε) und der minimalen Anzahl benachbarter Punkte zur Bildung eines Clusters (minPts). In dieser Arbeit wird ein Ansatz zur Automatisierung der Parameterabstimmung durch Integration der persistenten Homologie, einer Technik aus der topologischen Datenanalyse, vorgestellt. Die persistente Homologie analysiert topologische Merkmale, wie z. B. verbundene Komponenten und Zyklus, über mehrere räumliche Skalen hinweg und verbessert so die Genauigkeit und Robustheit der Clusterbildung.

Die Studie untersucht die Beziehung zwischen dichtebasiertem Clustering und topologischen Strukturen anhand synthetischer Datensätze mit unterschiedlicher Dimensionalität, Dichte und Noise-Anteile. Die Ergebnisse der Experimente zeigen, wie Einsichten aus der persistenten Homologie helfen, optimale Parameterwerte zu identifizieren, indem Clustering-Ergebnisse und die entsprechenden ε-Graphen in verschiedenen Einstellungen analysiert werden. Darüber hinaus wird die Robustheit der Methodik unter unterschiedlichen initialen Bedingungen und in Gegenwart von Rauschen validiert.

Die vorgeschlagene Technik verbessert den Prozess der Parameterwahl, sodass DBSCAN und verwandte Algorithmen effektiv auf verschiedenen Datensätzen angewendet werden können. Sie kombiniert topologische Erkenntnisse mit Clustering-Techniken und schafft eine Grundlage für robuste, skalierbare und automatisierte Ansätze zur Analyse komplexer Daten.

Contents

List of Tables

List of Figures

1 Introduction

Clustering is a valuable technique for organizing and understanding complex datasets by identifying natural groupings and patterns within the data [1, 2]. It is extensively researched and applied in various tasks such as customer segmentation in marketing, image segmentation and object recognition in computer vision, document clustering in natural language processing, communities identification in social network analysis, etc [2, 3]. In this context, density-based clustering methods emerge as an important clustering algorithm, offering advantages in identifying clusters of varying shapes and handling noise effectively.

Density-based clustering methods identify clusters by evaluating the local density of data points. The main idea is that clusters are areas with a higher density compared to the surrounding space. These methods detect clusters by searching for local maxima of density and grouping data points together based on these dense regions [4]. Essentially, density-based methods analyze an estimated probability density function to determine where the density of data exceeds a specific threshold. This threshold is a crucial parameter. If the threshold is set too low, separate clusters may merge into a single, larger cluster. On the other hand, if the threshold is set too high, clusters with lower densities may be missed. An optimal threshold will effectively distinguish and separate the original clusters. All these cases are illustrated in Figure 1.1 that is taken from [5].

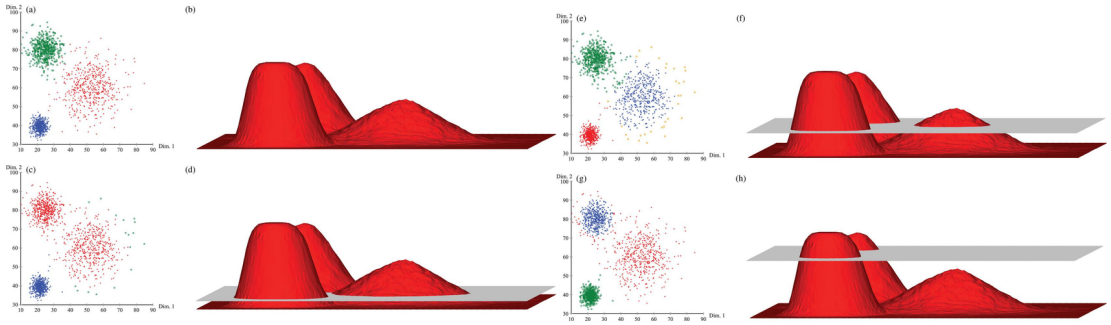

Figure 1.1: Density distributions of data points and clusters at various density levels: (a)-(b) the probability density function of the dataset, (c)-(d) the density level is too low, (e)-(f) the optimal density level that effectively separates the three original clusters, (g)-(h) the density level is too high. The picture is taken from [5]

Many density-based clustering methods are inspired by or based on the concepts introduced in Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [6]

and designed to address its limitations [7, 8, 9]. DBSCAN categorizes items based on their density-connectivity without requiring a predefined number of clusters. However, there are some user-defined parameter values, such as ε, which defines the distance for point neighborhood search, and minPts, the minimum neighboring points required to form a cluster around a given point.

While there are some methods available for choosing and calculating ε in DBSCAN, such as the elbow point method or using a k-distance graph [10], the parameter minPts remains less explored in terms of systematic study. MinPts plays a crucial role in determining the density of clusters and directly affects the algorithm's ability to differentiate between noise points and actual clusters. However, finding an optimal value for minPts is often a challenging task, as it depends on various factors such as the dataset's characteristics, the desired level of granularity in clustering, and the underlying distribution of data points. Despite its importance, there is no one-size-fits-all approach for choosing minPts, and it often requires experimentation and domain knowledge to determine a suitable value.

Persistent homology is a mathematical technique used in topological data analysis (TDA) [11] to study the shape and structure of data at different spatial scales, and facilitates the analysis of how topological characteristics like connected components, loops, and voids. In the context of DBSCAN, persistent homology can offer valuable insights into the choice of required parameters such as ε and minPts. Future research in this area could focus on developing automated techniques or guidelines for selecting minPts based on data properties or clustering objectives, thus enhancing the usability and effectiveness of density-based clustering algorithms (DBSCAN, HDBSCAN, DBSCAN-GM, etc) in diverse applications.

This research explores how integrating persistent homology with density-based clustering algorithms and parameter tuning for minPts enhances the algorithm's flexibility across various datasets. Merging TBA with clustering algorithms provides the way for automated parameter selection techniques. In conclusion, these enhancements contribute to improving the precision and robustness of clustering results in complex data analysis tasks.

2 Background

In recent years, significant progress has been made in the fields of data clustering and topological data analysis [6, 11, 12, 13, 14, 15], leading to the development of techniques such as DBSCAN, spectral clustering, and persistent homology. These methods share a common goal of uncovering underlying structures and patterns in complex datasets, despite their differences. The integration of these techniques (Fig. 2.1) can lead to a better understanding of data structures, especially in complex and high-dimensional datasets. This section provides an overview of their basic principles.

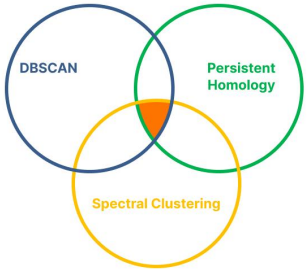

Figure 2.1: Combined techniques in clustering

2.1 DBSCAN

DBSCAN is a density-based clustering algorithm that can find varied shapes of varied size clusters [16] but struggles with clusters of varied density due to dependency on global input parameters. In the d-dimensional space, the limiting conditions include the threshold radius epsilon ε and the threshold size minPts. The sample points in the space are found in the iterative calculation, and the outlying data points are filtered to obtain the clustering results [17]. DBSCAN depends on some basic definitions as follows [6, 9, 17, 18, 7, 19]:

Definition 1 (ε-neighborhood) *The ε-neighborhood of a point p is defined as the set of points q within a distance of ε from p, denoted by $N_\varepsilon(p)$ and expressed by the formula:*

$$N_\varepsilon(p) = \{q \in D \mid Dist(p,q) \leq Eps\} \tag{1}$$

where $Dist(p,q)$ is the distance between point p and point q; $N_\varepsilon(p)$ includes all points in dataset D whose distance from points p is not larger than ε.

Definition 2 (Core Point) *For a dataset D the minimum neighborhood density threshold is set as minPts, if there is a point $p \in D$, and satisfies the formula (2), then the point p is a core object.*

$$|N_\varepsilon(p)| \geq MinPts \tag{2}$$

In formula (2), $|N_\varepsilon(p)|$ represents the number of points in the ε-neighborhood of point p. The core point is illustrated in Figure 2.2.

Definition 3 (Directly Density-reachable) *For a dataset D, if there exists a point q in the ε-neighborhood of point p ($q \in N_\varepsilon(p)$) and the point p is a core point, then point q is Directly Density-reachable from the point p.*

Definition 4 (Border Point) *For a dataset D, a point $p \in D$ is a border point if it satisfies the following criteria:*

1. $p \in N_\varepsilon(q)$
2. $|N_\varepsilon(p)| < minPts$

In other words, a border point has fewer than $minPts$ points within its ε-neighborhood, but it is directly density-reachable from at least one core point. The border point is illustrated in Figure 2.2.

Definition 5 (Density-Reachable) *For a dataset D, if there exists a sequence of points $p_1, p_2, ..., p_n \in D$, for each p_i (where $0 < i < n$), p_{i+1} is density-reachable from point p_i then point p_n is density-reachable from point p_1. Density reachability is asymmetric.*

Definition 6 (Density-connected) *For a dataset D, if there exists a point $o \in D$ such that points p and q are both density-reachable from point o, then points p and q are density-connected. The density connectivity is symmetric.*

Definition 7 (Cluster) *For a dataset D, the cluster C is a non-empty subset of the dataset D satisfying the following criteria:*

1. $\forall p, q$: *if $p \in C$ and q is density-reachable from p, then $q \in C$.*
2. $\forall p, q \in C : p$ *is density-connected to q.*

Definition 8 (Noise point) *For a dataset D, if the point p does not belong to any cluster, then the point p is a noise point:*

$$noise = \{p \in D \mid \forall i : p \notin C_i\} \tag{3}$$

The noise point is illustrated in Figure 2.2.

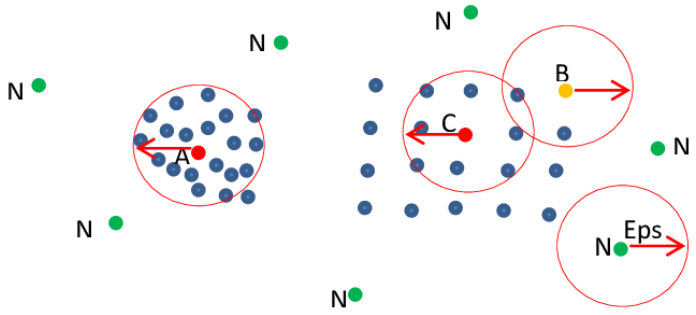

Figure 2.2: DBSCAN clustering process: points A and C are core points, B is a border point, points labeled N are noise points, and the red circles represent ε-neighborhood of the corresponding point, minPts = 4. The figure is taken from [16].

Definition 9 (Core Distance) *The core distance of a point p in a dataset D is defined as the distance from p to its k-th nearest neighbor [20, 21], where the value of k includes the point p itself and expressed as:*

$$d_{core}(p) = d(p, p_k), \tag{4}$$

where p_k is the k-th nearest neighbor of point p. The core distance is illustrated in Figure 2.3.

Definition 10 (Mutual Reachability Distance) *The mutual reachability distance between two points p and q in a dataset D is the maximum value of: the core distance of p, the core distance of q, or the distance between p and q [21]. It is expressed as:*

$$d_r(p, q) = \max\left(d_{core}(p), d_{core}(q), d(p, q)\right), \tag{5}$$

where $d(p, q)$ usually represents the Euclidean distance between p and q [20]. The mutual reachability distance is illustrated in Figure 2.3.

Definition 11 (Density-Connectivity Distance (dc-dist)) *The density-connectivity distance between two points p and q in a dataset D is the minimax distance along the path connecting p and q through a sequence of intermediate points. This path is based on the mutual reachability distance, ensuring that both density and connectivity are considered [20]. The density-connectivity distance is given by:*

$$d_{dc}(p, q) = \min\max\left(d_r(p, p_1), d_r(p_1, p_2), \ldots, d_r(p_n, q)\right), \tag{6}$$

where the sequence of points $p_1, p_2, \ldots, p_n$ connects p to q.

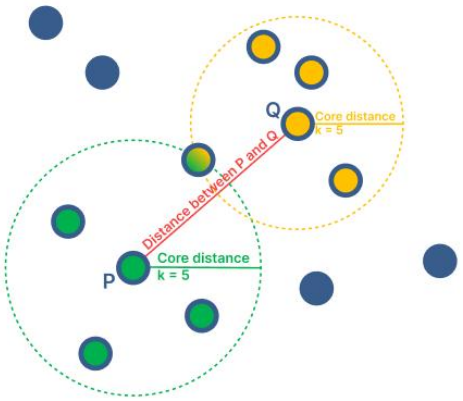

Figure 2.3: Core and mutual reachability distances in clustering. The core distances for the points P and Q when k = 5. Mutual reachability distance = the distance between the points P and Q in this case. Inspired by [21].

2.2 Spectral Clustering

Spectral clustering is a powerful technique for clustering data that identifies clusters with various shapes and complex structures in high-dimensional datasets. In comparison with DBSCAN, spectral clustering works with a graph representation of the data, focusing on the similarity between data points. DBSCAN identifies clusters as connected components in an ε-graph based on density. Spectral clustering also detects these connected components using the eigenvectors of the graph Laplacian matrix. [22].

Datasets can be represented as graphs, where data points are treated as nodes and the relationships between them as edges. Spectral clustering, as a graph-based technique, transforms data into such a graph representation where nodes represent data points and edges denote pairwise similarities between them [23].

Definition 12 (Graph) *An undirected graph $G = (V, E)$ is defined as a set of nodes V and a set of edges E, where E is some subset of the pairs $\{\{u, v\} : u, v \in V, u \neq v\}$.* [24]

In this context, each data point becomes a node in the graph, and the edges between the nodes reflect how similar or connected the data points are based on the affinity matrix. This matrix quantifies the strength of these connections between points.

The graph's structure, including concepts like nodes, edges, and cycles, forms the foundation for spectral clustering. To create this graph-based representation of the data, an affinity matrix is constructed. This matrix captures the pairwise relationships between the data points by quantifying the strength of the connections between them. The core process in spectral clustering is the eigenvalue decomposition of the graph Laplacian matrix, generating eigenvectors that encode clustering information. Clusters are then identified by analyzing spectral embeddings, often using techniques like spectral partitioning based on the graph's cut ratio [23].

One of the key advantages of spectral clustering is its capability to handle complex shapes, which are common in real-world data. By focusing on the pairwise relationships between data points rather than their explicit geometric properties, spectral clustering can identify clusters with arbitrary shapes and capture intricate relationships that may not be apparent in the original feature space. The graph representation of data enables spectral clustering to analyze high-dimensional datasets effectively. This technique often applies in image segmentation to group pixels into meaningful regions based on their similarity [14].

2.3 Persistent Homology

Persistent homology is a TDA method that helps to understand the shape and connectivity of data on different scales. To find the persistent homology of a space, the space must first be represented as a simplicial complex [12, 13].

Definition 13 (Simplicial complex) *A simplicial complex is a geometric construct consisting of vertices, edges, triangles, and their higher-dimensional analogs (Figure 2.4). These complexes are formed from the dataset and serve as a topological representation by combining simpler components through combinatorial rules.*

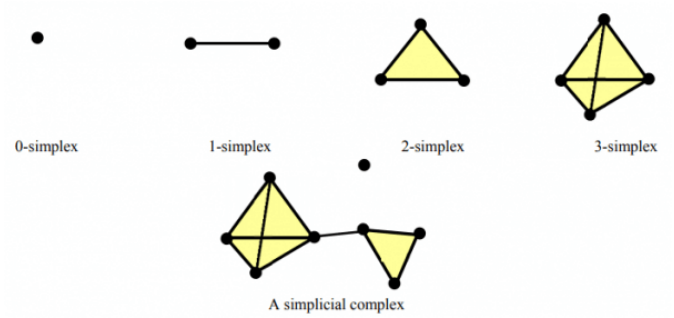

Figure 2.4: The 0-simplices represent vertices or points, 1-simplices represent edges or lines, 2-simplices represent triangles and 3-simplices represent tetrahedra. A simplicial complex is built by a combination of these simplices. The picture is taken from [25].

This complex is created by establishing a distance function on the underlying space, which in turn defines a filtration of the simplicial complex [12, 13]. The process is illustrated on Figure 2.5.

Definition 14 (Filtration) *A filtration is a nested sequence of simplicial complexes, each a subset of the next, with different types defined by their nesting rules.*

Persistent homology utilizes the filtration of a simplicial complex to track the evolution of homological features across different levels, enabling the identification of persistent features that are indicative of meaningful structural characteristics in the data [11].

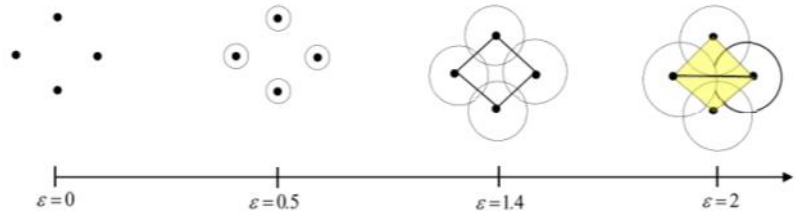

Figure 2.5: The formation of simplicial complexes based on the values of the filtration ε. The picture is taken from [25].

In this work, up to 2-simplices are tracked in the datasets. The simplicial complex that consists only of nodes (0-simplicas) and paths of edges (1-simplices) is called a connected component. The cycles of length 3 are tracked as 2-simplices.

Definition 15 (Connected component) *Given a graph $G = (V, E)$, a connected component is a maximal subset of vertices $V' \subseteq V$ such that for any two vertices $v_i, v_j \in V'$, there exists a path of edges $\{e_1, e_2, \ldots, e_k\} \subseteq E$ connecting v_i to v_j.*

Definition 16 (Cycle of length k) *A cycle of length k in G is a sequence of nodes $v_1, v_2, \ldots, v_{k+1}$ in V such that $\{v_i, v_{i+1}\} \in E$ for all $i \in \{1, \ldots, k\}$. Equivalently, it can be described by the corresponding sequence of k edges. Some examples are shown in Figure 2.6*

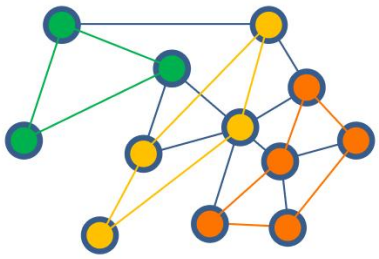

Figure 2.6: The illustration of length of the cycles in the graph. The cycle of length 3 is green. The cycle of length 4 is yellow. The cycle of length 5 is orange.

3 Related Work

Density-based clustering includes a range of algorithms that detect clusters by analyzing the density of data points within a dataset. DBSCAN is one of the most robust methods for density-based clustering, which groups points based on their density, isolating outliers effectively. However, determining the optimal parameters like ε and minPts for DBSCAN can be challenging. Various techniques, such as the Elbow Method [26], and algorithms llike OPTICS [7], DBSCAN-GM [9], CoExDBSCAN [27], and HDBSCAN [15], provide methodologies to estimate these parameters more effectively.

One of the techniques for determining a suitable ε is the k-distance graph, which involves plotting the distances of each data point to its k-nearest neighbor against various values of k. By observing this plot, the goal is to pinpoint an "elbow" point where the rate of change sharply decreases. The elbow indicates a potential optimal ε value, distinguishing core points from outliers [26, 19]. The method is simple to implement and easy to understand, making it accessible to researchers and practitioners in the field. However, it is important to note that the Elbow Method relies on subjective interpretation, which can sometimes lead to ambiguity, especially in datasets with complex density structures or high-dimensional, noisy data. Despite this limitation, its visual guidance remains valuable, often complemented by other techniques and domain knowledge for more accurate parameter selection in clustering algorithms.

Grid search with cross-validation is another commonly applied method to estimate DBSCAN parameters [4]. It systematically explores various combinations of ε and minPts, selecting the pair that gives the best clustering results. Grid search is effective but it is also computationally expensive, especially for large datasets. Grid search does not provide any deeper understanding of density and relies on external validation metrics [28].

Further enhancing DBSCAN's parameter tuning, OPTICS [7] offers a nuanced approach that provides a more detailed clustering structure by ordering points based on their reachability distance. This distance measure considers not just how far points are from each other but also the density of points in between. In the context of DBSCAN, OPTICS can assist in determining suitable values for ε and minPts. The reachability plot generated by OPTICS provides insights into how points are connected and the density transitions within the dataset. One advantage of OPTICS is its ability to handle varying densities. This is crucial for DBSCAN, as it allows the algorithm to adapt to clusters of different densities without compromising accuracy. Additionally, OPTICS offers a hierarchical view of clustering, which can be beneficial when exploring data sets with nested clusters or varying cluster sizes. However, like any algorithm, OPTICS has its challenges. It can be computationally intensive, especially for large datasets, and choosing the right parameters can require careful consideration.

The idea of DBSCAN-GM [9] is to set the initial DBSCAN parameter by exploring the

advantages of Gaussian Means. During the initial phase of DBSCAN-GM, Gaussian Means is employed to create several seed clusters without any prior information. These identified clusters, along with their centers, provide a reliable estimation for the parameters ε and minPts, as these parameters are influenced by the number of points and their distributions within each cluster.

HDBSCAN represents another significant advancement in clustering algorithms [15]. HDBSCAN simplifies parameterization by using a minimum cluster size to control the granularity of hierarchical clustering. Unlike DBSCAN, HDBSCAN does not require users to specify ε directly, offering a more intuitive and adaptable clustering approach. This hierarchical approach to clustering allows for a more detailed and nuanced representation of the data's clustering structure, providing insights into both global and local cluster structures within the dataset.

Progress in parameter estimation techniques has significantly improved the effectiveness of density-based clustering algorithms. However, it is essential to understand the stability and persistence of clusters across various parameter settings to achieve robust clustering results. Persistent homology provides a unique perspective on the stability and longevity of clusters. By tracking topological features at different scales or parameter values, this method can provide insights into the robustness and significance of clusters.

In recent years, persistent homology has been explored in the fields of clustering and graph theory [11, 12, 13, 29, 30]. This technique stands out as a powerful tool in TDA, allowing the extraction of crucial topological features from complex data sets and facilitating a comprehensive study of their underlying structures. The papers delve into a variety of topics related to persistent homology, covering its mathematical foundations, optimization methods, visualization techniques, and real-world applications. Through these discussions, a better understanding of the potential and capabilities of persistent homology emerges, highlighting its significance in advanced data analysis.

One of the relevant studies [29] is the development and application of the Persistable algorithm, a density-based clustering method that combines hierarchical clustering with a persistence-based flattening technique. This approach aims to extract clusters from complex datasets at multiple scales. The stability and consistency of the algorithm contribute to reliable clustering results. Additionally, the paper explores the stability and consistency of clustering algorithms and presents the kernel linkage method as a stable and consistent approach for density-based clustering. The Persistable approach shows promise in identifying multiscale cluster structures and outperforms other algorithms on challenging datasets. Its scalability for the number of data points and the density threshold parameter enables fine clusterings in large datasets.

A new method for constructing a filtration in cluster analysis using persistent homology within weighted graphs has been developed [13]. This new approach, known as the k-cluster filtration, delays the inclusion of vertices until they become part of adequately large clusters, leading to more informative persistence diagrams. Researchers have also investigated the universality properties associated with persistence values in the k-cluster filtration. On the other hand, adapting the k-means algorithm to persistence diagrams is investigated using distance metrics [12]. Delving into different representations of these

diagrams and examining how the k-means algorithm converges in this context provides a structured analysis that enhances effectively using k-means clustering, especially in situations where traditional metrics face challenges.

4 Improving Density-Based Clustering With Topological Data Analysis

The integration of persistent homology with density-based clustering, particularly DBSCAN, offers promising opportunities to enhance the robustness and accuracy of clustering results. DBSCAN is a highly effective algorithm for identifying clusters of arbitrary shapes and handling noise. However, it is heavily influenced by two key parameters: ε (the radius for the neighborhood search) and minPts (the minimum neighboring points required to form a cluster) which are described in Section 2. Finding optimal parameter settings has always been challenging, frequently dependent on experimentation or intuitive techniques.

Persistent homology, a key technique in TDA, offers a structured, mathematical approach to understanding data structure across different parameter settings. By analyzing the data's topological features, such as the number of connected components and loops, we can identify optimal configurations for the clustering algorithm that can enhance its robustness.

This research aims to explore the relationship between density-based clustering and persistent homology. The primary goal is to identify the optimal values for the two critical parameters of DBSCAN using the information about data structure. Through a systematic analysis of synthetic datasets, this study investigates how varying these parameters influences the clustering results and reveals structural patterns in the data.

Thus, this study focuses on three core research questions:

1. What insights can be gained by integrating persistent homology with DBSCAN?
2. How can these insights help in identifying optimal parameter settings for DBSCAN?
3. How robust are DBSCAN's clustering results to variations in its initial parameters?

4.1 Integration of Persistent Homology and DBSCAN

DBSCAN is a density-based clustering algorithm that groups data points based on their proximity to each other as is described in Section 2.1. While this method is effective in certain scenarios, it may struggle to adapt to varying density levels and complex data structures, especially in high-dimensional spaces. Traditional approaches to tuning the DBSCAN parameters may not generalize well across different datasets.

On the other hand, persistent homology provides a topological perspective that can detect deeper structural patterns in the data. This technique can identify the stability of clusters (connected components) and loops. By offering a more comprehensive view of the data, persistent homology can lead to more accurate and reliable clustering results.

Persistent homology works by constructing a simplicial complex and monitoring how its characteristics change as the topological space of the dataset evolves as it is described in Section 2.3. For example, this evolution can be observed when new edges are added to the DBSCAN graph representation of the data. This method enables us to detect significant transitions, such as the merging of two clusters or the emergence of new topological structures, like loops.

Systematically tracking the evolution of topological features as ε and minPts vary persistent homology allows to identify stable connected components (clusters) that persist across a range of parameter settings, detect loops that indicate complex relationships between data points.

By observing topological changes in DBSCAN clustering results, persistent homology can provide insights into how parameterization affects clustering outcomes, such as merging clusters or the emergence of significant loops. This understanding allows us to hypothesize that persistent homology can be potentially used to determine optimal values for the parameters ε and minPts. This process identifies stable topological structures, thus providing a mathematically sound method for improving clustering results. The analysis of topological features that is extracted from experimental datasets is detailed in Section 5.3.

4.2 Topological Insights for Identifying Optimal Clustering Parameters

A key advantage of persistent homology is its ability to evaluate the stability of topological features across different scales. For clustering, this means to identifying parameter values that yield robust and meaningful results. As the neighborhood radius ε and the minPts threshold change, the connected components and loops of the dataset fluctuate. The persistence of these features, especially across a broad range of parameter values, can indicate stability in the clustering structure.

For instance, stable connected components that persist over a wide range of ε values are likely to represent true number of clusters within the data, rather than noise. Similarly, the presence of loops in the data can indicate more complex relationships between clusters, such as overlapping or nested regions.

Optimal clustering configurations, evaluated using external validation metrics will correspond to observed patterns in the persistence of connected components and loops within the topological space. Specifically, the number of connected components is expected to stabilize at the correct number of clusters as the parameters ε and minPts approach their optimal values. Additionally, the loop count is expected to demonstrate identifiable patterns. The experiments validating these expectations are detailed in Section 5.2 - 5.3

4.3 Robustness of Clustering Parameters

The parameters ε and minPts in DBSCAN are crucial for shaping the clusters the algorithm identifies as mentioned before. Increasing the ε value expands the neighborhood radius, which adds more edges to the connectivity graph, potentially causing clusters to merge and new topological features, like loops, to form. Likewise, adjusting minPts alters the threshold for how densely connected a set of points must be to constitute a cluster. By systematically tracking these changes through persistent homology, we gain valuable insights into how different parameter choices influence the data's underlying structure.

We assume that variations in ε and minPts influence the formation and persistence of connected components and loops, with stable features emerging at parameter values that lead to optimal clustering results. To explore this hypothesis, ε will be varied over a broad range to observe changes in the number of connected components and loops and minPts will be adjusted to assess its impact on the density and structure of the clusters that form. The details of these experiments are provided in Section 5.4.

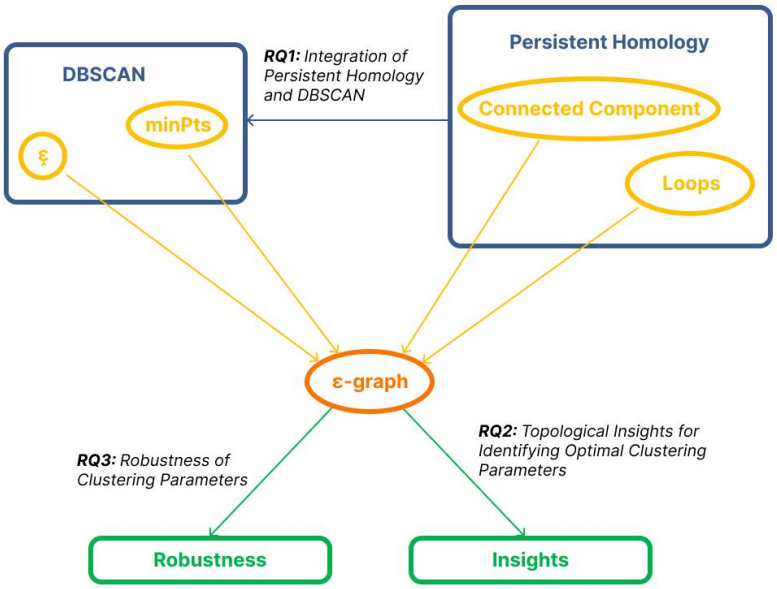

Figure 4.1: Schema of the research questions

5 Experiments

This section provides a detailed description of the experiments conducted to investigate how integration of persistent homology can improve the DBSCAN algorithm. Using the datasets discussed in this section, we address the research questions presented in Section 4, offering insights into the potential benefits of this combination of clustering technique and TDA method.

5.1 Data

This study uses the DENSIRED data generation tool to create synthetic datasets where data points are certainly density-connected [31]. The tool generates datasets with multiple clusters, each characterized by a high density of data points. The clusters exhibit varying degrees of complexity in their shape. These datasets served as the basis for graph construction and further analysis using persistent homology, to determine optimal parameters for DBSCAN.

The experiments were conducted using datasets with different dimensions (d) ranging from 30 to 100 and numbers of samples (n) varying from 1000 to 5000. Specifically, $d = [30, 50, 100]$ and $n = [1000, 1500, 2000, 2500, 3000]$. The number of clusters (clunum) in these datasets was set between 5 and 30, for some datasets noise levels were set up at 1-3%, simulating outliers, and background noise. Additionally, datasets were generated with varying density parameters such as step (δ), momentum (ω), branch (β) and star (χ). The step parameter defines the base distance between cluster cores, influencing the overall spread of the clusters in the feature space. The momentum factor assigns a global value to all clusters, determining their "stickiness", or the tendency of data points to remain close to their originating core. In this case, momentum was set relatively high to ensure clusters maintain cohesion. The branch parameter controls the probability of a cluster branching off from its initial structure, and for this study, it was set to a low value to limit the complexity of cluster formations. The star parameter influences the likelihood that the initial core will be chosen when the clustering process is restarted. In this case, it has been set to a low value to ensure that the remaining cores have an equal probability of being selected for subsequent iterations. These variations allowed for the analysis of different dataset structures and complexities.

In the study, the experiments are grouped based on different parameter settings. Each group contains up to 10 experiments and varies by specific configurations of the dimensions, number of samples, number of clusters, step, momentum, branch, star and noise level parameters, allowing for a comprehensive exploration of how different density dynamics influence clustering behavior. An analysis is conducted on the average values of key

metrics across all experiments within each group to ensure valuable and meaningful results. The exact values of the parameters set for the experiment groups are presented in Table 5.1, including the group with parameter settings from [31]. Additionally, Figure 5.1 illustrates the $2d$-projection of the datasets from each group using Principal Component Analysis (PCA).

№	dim	n	clunum	δ	ω	β	χ	noise level
1	50	1000	5	1	0.9	0.05	0	0
2	50	1500	10	1	0.8	0.05	0	0.01
3	30	1000	12	1	0.9	0.1	0	0
4	50	2000	15	1	0.8	0.05	0	0
5	50	2500	20	1	0.9	0.05	0	0.03
6	50	3000	30	1	0.95	0	0	0

Table 5.1: Dataset settings

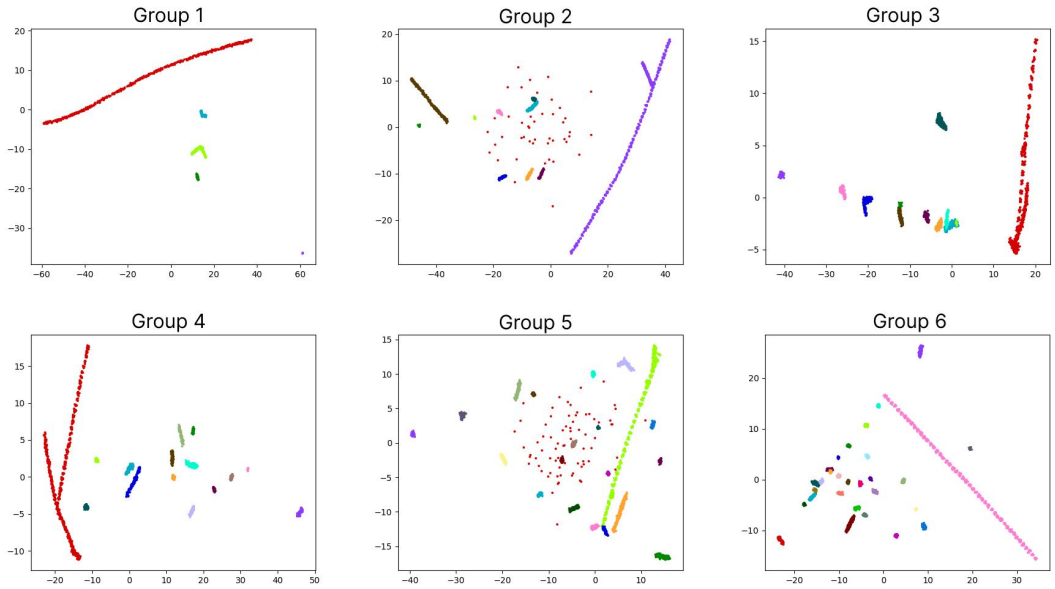

Figure 5.1: Generated datasets

5.2 Parameter Selection and Topological Feature Extraction

To improve the density-based algorithm DBSCAN two crucial parameters must be carefully selected. The first one is ε and it defines the ε-neighborhood of each point, the other one is minPts which determines the minimum number of points required to create a cluster as is described in Section 2.1. This study considers minPts ranging from 2 to 20 and

computes ε as the distances between every pair of points. The dc-distance metric that is defined in Section 2.1 (Definition 11) is employed to measure the distance between data points.

For each specific combination of the minPts and ε parameters, a ε-neighborhood graph is constructed. This graph provides a visual representation of the data points and their relationships based on the given density parameters. In this graph, each data point is represented as a node. An edge is drawn between two nodes (data points) if the dc-distance between them is less than the specified ε threshold. Essentially, an edge indicates that two points are considered neighbors within the defined density range.

Persistent homology is used to extract valuable topological information from the constructed graphs. This powerful technique allows to identify and track topological features as the observation scale varies. Specifically, the study is focused on the number of connected components that quantify the number of distinct clusters present in the data and the number of loops that provide insights into the complexity of the cluster structures. To gain deeper insights into the clustering process, the labels assigned to each data point are saved for every parameter pair. The analysis of these labels allows us to identify patterns in point assignments and cluster formations.

The computed topological features serve as valuable indicators for assessing the quality of clustering results. Analyzing these characteristics, parameter settings that result in persistence and meaningful cluster structures can be found. The goal is to select parameter pairs that produce stable topological features across different noise levels and data distributions. To support this process, visualization techniques are used to explore the relationship between parameter settings and topological features.

5.3 Analysis Extracted Topological Feature

Optimizing density-based clustering algorithms requires understanding the relationship between parameter selection and extracted topological features. This section explores the analysis of simplicial complexes (Definition 13) obtained from the ε-graphs in DBSCAN, focusing on characteristics that can lead to appropriate parameter selection.

The dependence of the extracted topological feature values and initial parameters of the DBSCAN algorithm is illustrated in the heatmap (Figures 5.2 and 5.3). The x-axis corresponds to ε, and the y-axis represents the minPts values. The first column of Figure 5.2 displays the complete array of connected components for all possible combinations of DBSCAN parameters. The second column provides detailed slices of the optimal parameter settings. Analyzing Figure 5.2 for the number of connected components in different data groups, we can see that the "correct" result typically lies within a specific triangular region. When the ε and minPts values are low, the algorithm detects a large number of clusters. It is illustrated by the yellow area on the heatmap, where the cluster count significantly exceeds the actual value, for example, if the right number of clusters is 5, the ε-graph can consist of more than 150 connected components for such parameter pairs. However, as ε increases, the number of clusters stabilizes, represented by the darker purple regions on the heatmap, indicating closer alignment with the true number of

clusters. The red areas highlight the combinations of parameters that produce the most accurate clustering results.

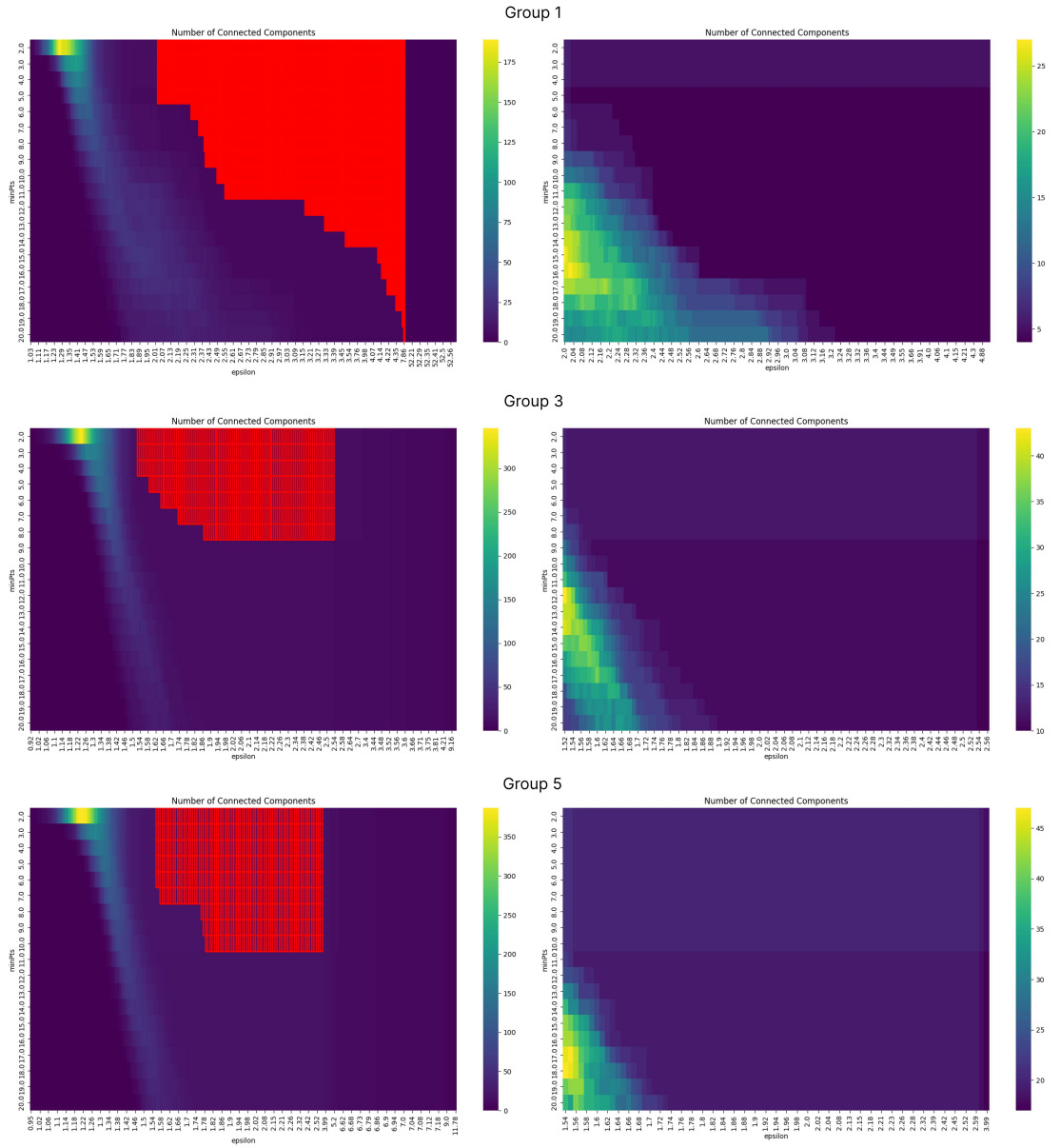

Figure 5.2: Heatmaps of the number of connected components for groups 1, 3 and 5. The x-axis corresponds to ε and the y-axis – to minPts. The left column displays the complete array of connected components for all combinations of ε and minPts. The right column provides detailed slices of the optimal parameter settings.

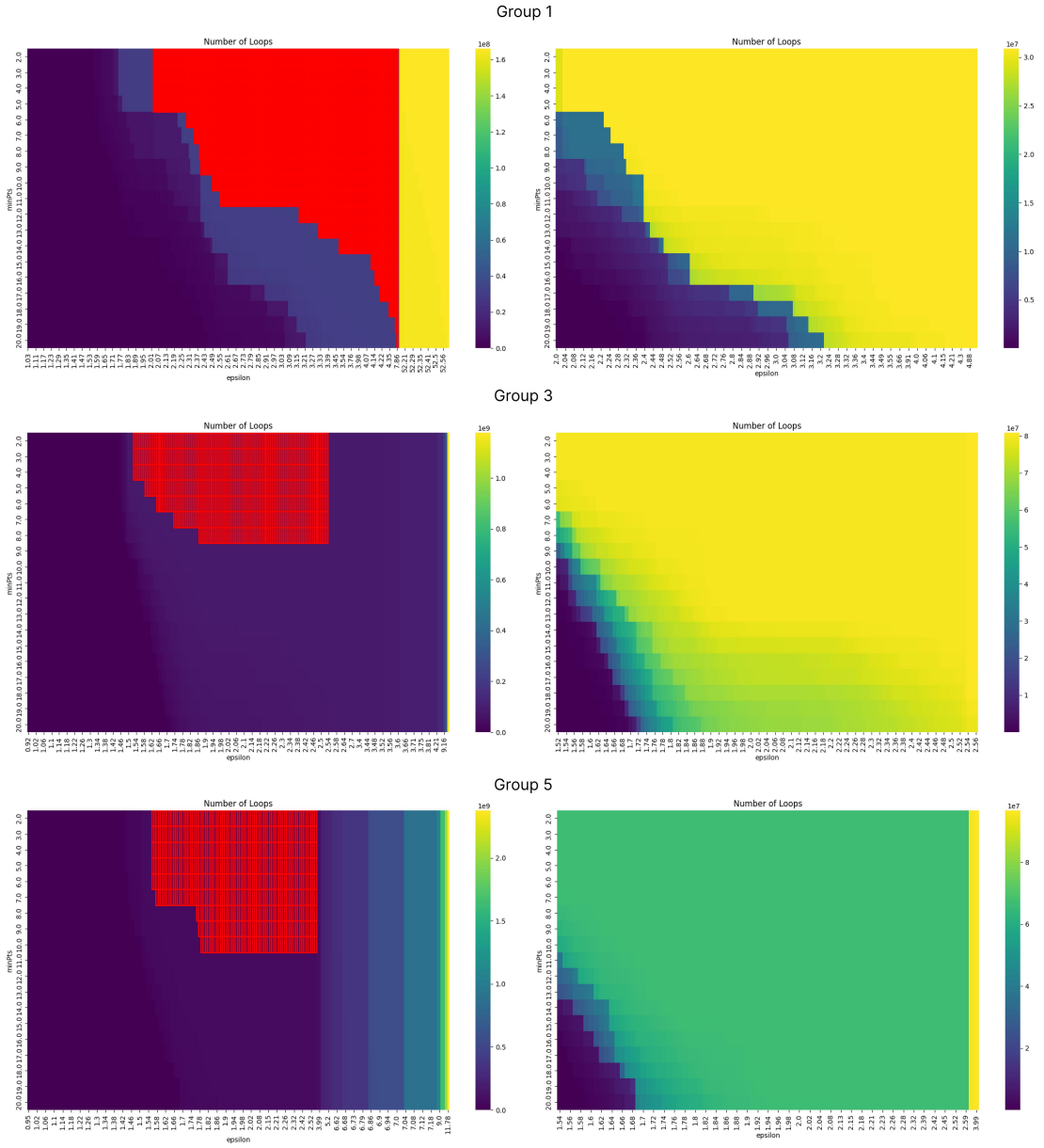

Figure 5.3: Heatmaps of the number of loops for groups 1, 3 and 5. The x-axis corresponds to ε and the y-axis – to minPts. The left column displays the complete array of loops for all combinations of ε and minPts. The right column provides detailed slices of the optimal parameter settings.

The other heatmap in Figure 5.3 shows the number of loops in the ε-graph. Similarly to Figure 5.2, the heatmaps in the first column display data for all combinations of minPts and ε, while the second column shows slices corresponding to the optimal parameter

settings. As ε increases, the number of loops also rises, indicating that specific ranges of the number of loops correspond to optimal parameter settings.

These heatmaps help to identify suitable parameter settings when the correct number of clusters is known in advance. However, in practice, the DBSCAN algorithm is often used without prior knowledge of the precise number of clusters. To address this challenge, a more detailed analysis of the internal characteristics of the ε-graph is proposed. This analysis aims to explore how these characteristics can effectively guide the selection of the minPts and ε parameters, even if the actual number of clusters is unknown.

5.3.1 The Number of Connected Components

Analysis of the number of connected components in the constructed ε-graph provides key insights into clustering parameter selection and dataset characteristics. Histograms of the frequency of the number of connected components provide information on the distribution of potential clusters across different parameter combinations (Figure 5.4), accounting for the presence of noise in some datasets (Figures 5.5 and 5.6).

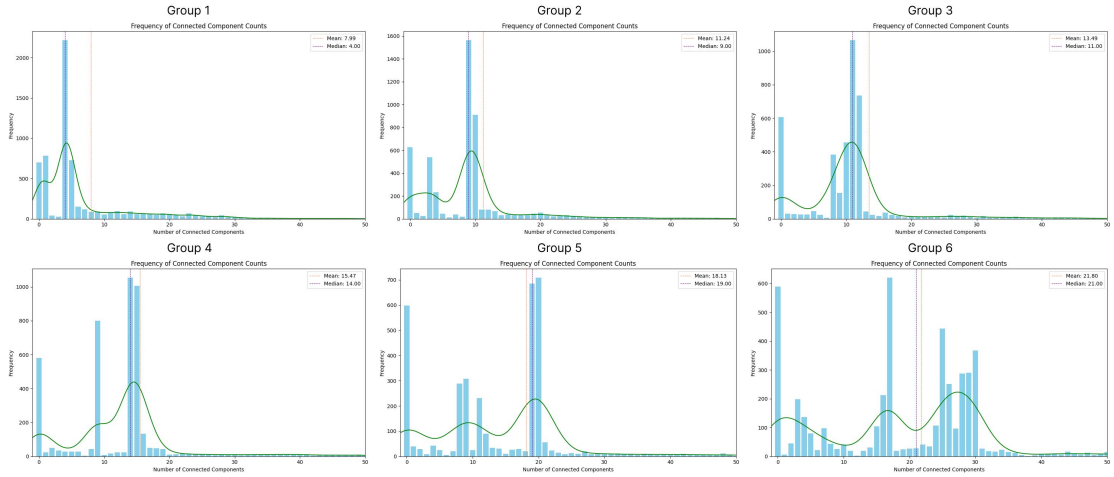

Figure 5.4: Frequency of connected components for generated datasets. The x-axis represents the number of connected components and the y-axis – the frequency of appearance the number when the ε-graph is contracted for different combinations of DBSCAN initial parameters.

The true number of clusters appears among the top 5 peaks of the histogram in 90% of the experiments, which can help in evaluating the data structure. Generally, the bins of the histogram centered around significant values that indicate the most suitable number of clusters for the dataset. For example, as shown in Figure 5.4, when clustering group 3, the most common number of clusters is 11, but 12 is also a significant value ranking second in frequency, even though it is almost twice as infrequent. This may mean the following:

- one of the clusters is tiny and often classified as noise;
- two clusters are too close to each other to be distinguished at certain values of ε and minPts;
- two clusters are small enough to merge into a single cluster at certain minPts values.

According to the histogram for group 4 in Figure 5.4, the optimal number of clusters could be 14 or 15, as both occur with similar frequency. Based on the data structure model for this group (Figure 5.1), it can be assumed that the red cluster is occasionally split into two different clusters.

Dataset 6 originally consisted of 3000 points grouped into 30 clusters. In the histogram for group 6 (Figure 5.4) the number of connected components is more widely distributed between 0 and 30 compared to other groups. Nevertheless, the true number of clusters is still among the top 3 most frequent peaks. A common peak appears at 0, which can be ignored during analysis, as it indicates that all points are identified as noise due to an irrelevant combination of ε and minPts. This behavior is observed across all datasets.

When noise (1-3%) is initially present in the dataset, a distinct peak in the histogram often occurs at 1 connected component. This phenomenon indicates that all data points merge into a single cluster because noise points bridge several clusters. Despite this, the sequence of significant peaks often corresponds to the correct number of clusters. In Figures 5.5 and 5.6, histograms for noisy datasets demonstrate that while the "correct" number of clusters is less frequent than in clean datasets, the peaks are still clear. These experiments highlight the stability of this histogram-based method, even in the presence of noise.

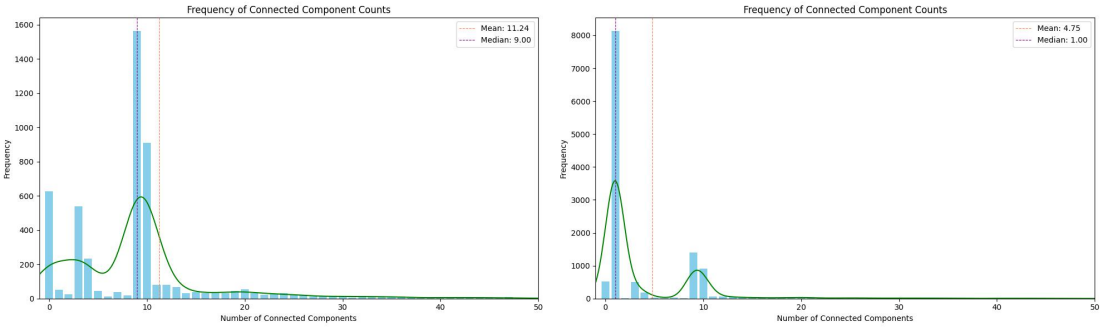

Figure 5.5: Frequency of connected components for group 2 with 1% noise. The left histogram presents the frequency of connected components in clean data. The right histogram presents the same in noisy data.

These findings provide a robust framework for interpreting histogram results, especially in datasets where noise may obscure the true clustering structure. Through careful analysis, not only can the number of clusters be detected, but also the optimal combination of

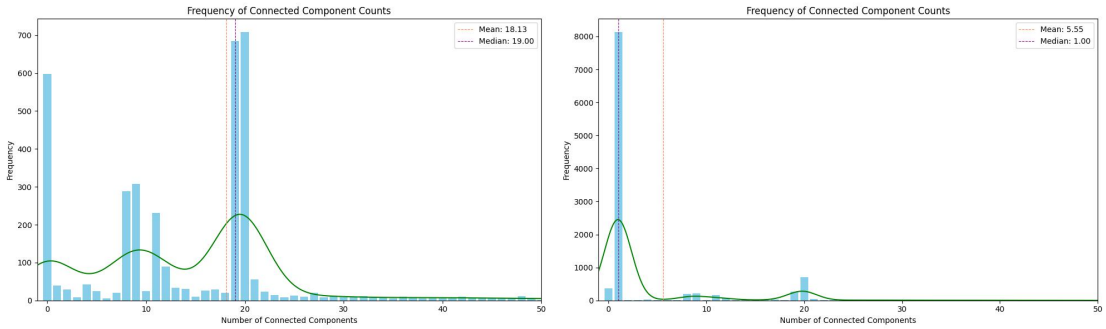

Figure 5.6: Frequency of connected components for group 5 with 3% noise. The left histogram presents the frequency of connected components in clean data. The right histogram presents the same in noisy data.

DBSCAN parameters ε and minPts can be determined. This ensures meaningful and realistic clustering results for the given data.

5.3.2 The Number of Loops

To improve the analysis, an additional parameter of persistent homology, the number of loops in the constructed ε-graph, was analyzed. This parameter estimates the topological complexity of the data. For each ε-graph built for a unique combination of minPts and ε, the number of cycles length 3 (Definition 16) was calculated. For each value of the number of clusters, all possible values of the number of loops and their frequencies were considered. This resulted in a series of histograms, each showing the distribution of the number of loops on graphs with the same number of connected components (Figure 5.7-5.11).

Analyzing these histograms for experimental data groups found several meaningful patterns. The number of cycles corresponding to the most common values of connected components tends to concentrate within a specific range. According to Figure 5.7, in group 3, the number of loops for the most frequent number of connected components lies within the range of 0 to $1 \cdot 10^8$. The more detailed illustration of the precise range for the number of loops, shown in Figure 5.8, indicates that for the most frequently occurring connected components, the number of loops is approximately between $0.7 \cdot 10^8$ and $0.8 \cdot 10^8$.

Similarly, for datasets from group 5, this measure is concentrated between $0.6 \cdot 10^8$ and $0.7 \cdot 10^8$ cycles (Figure 5.9 and 5.10). Moreover, this measurement is valid only for the true number of clusters. The most frequent numbers of connected components in group 5 are 20, 19, 0, 9, 8, 11, and 12, as shown in the frequency histogram in Figure 5.4. To estimate the suitable number of clusters, three potential groups can be considered: 20 and 19, 9 and 8, and 11 and 12. However, the last two groups could be dismissed based on the histogram of the number of loops in Figures 5.9 and 5.10. The number of loops for these values is concentrated in different ranges. When the number of connected components is

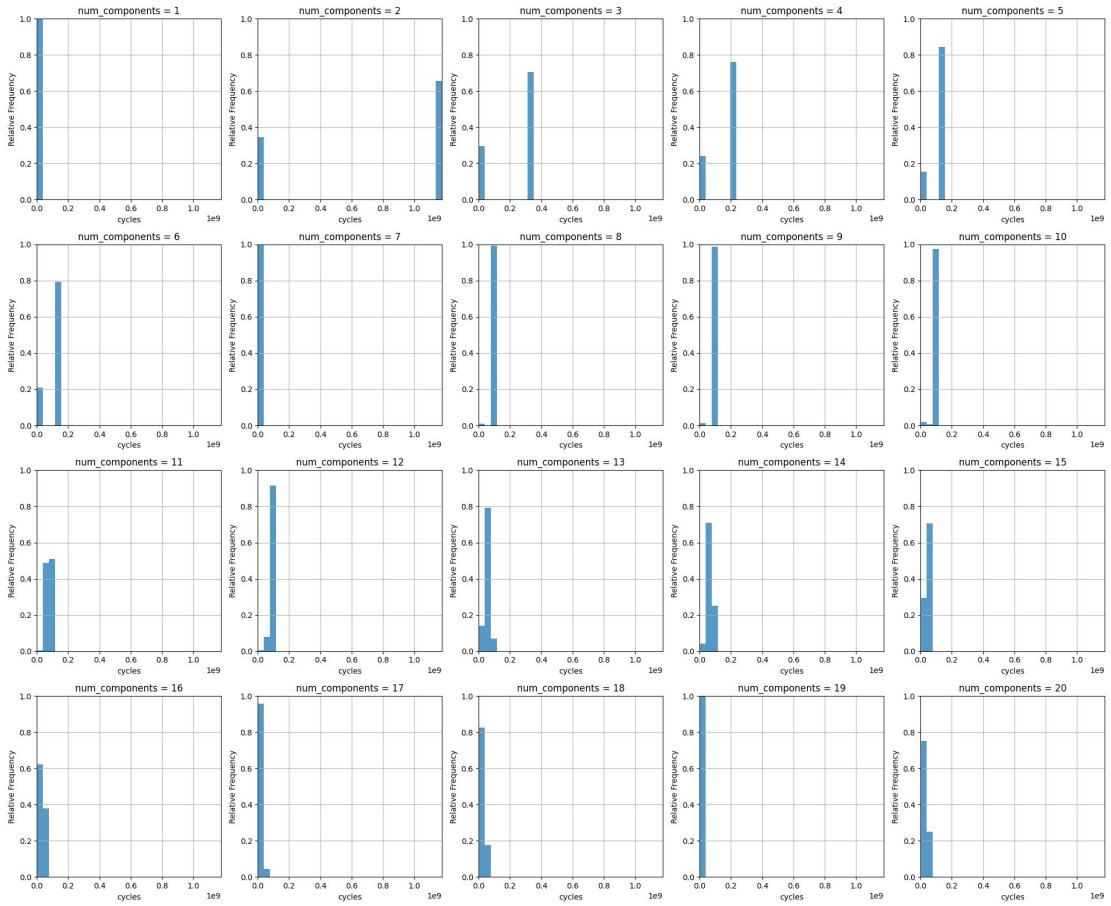

Figure 5.7: Histograms of the original number of loops for group 3. Each subplot corresponds to a specific number of connected components (num_components). The x-axis represents the number of cycles, and the y-axis shows the relative frequency.

8 the number of loops is mostly around $10 \cdot 10^8$. For the 9 connected components, the number of loops is around $6 \cdot 10^8$. For the 11 and 12 connected components, the number of loops is between $3 \cdot 10^8$ and $4 \cdot 10^8$. For more detailed analysis, the number of loops until $1 \cdot 10^8$ is illustrated more precise in Figure 5.10. The number of loops still concentrated between $0.6 \cdot 10^8$ and $0.7 \cdot 10^8$ cycles. The relative frequency of this value is greater than 0.9.

Based on these findings, it can be concluded that the number of cycles corresponding to the correct value of the clusters lies within a specific interval. However, further research is necessary to develop a reliable method for determining this range. Such advancements would significantly improve the accuracy of the cluster estimation by

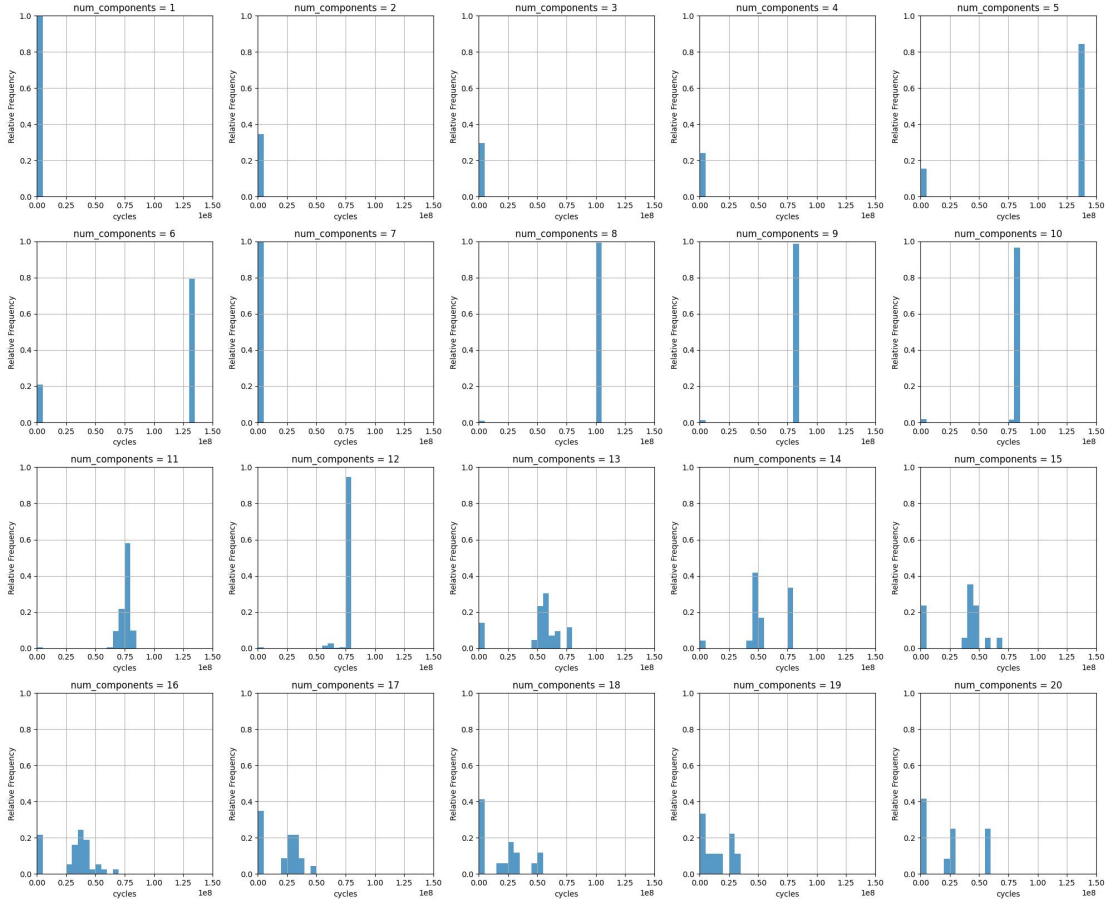

Figure 5.8: Histograms of the original number of loops for group 3 with range up to $1.5 \cdot 10^8$. Each subplot corresponds to a specific number of connected components (num_components). The x-axis represents the number of cycles, and the y-axis shows the relative frequency.

filtering out prominent peaks in the histogram of the frequencies of the connected components (Figure 5.4). Furthermore, the presence of similar patterns across different groups suggests potential variations in the distribution of data points. The concentration of cycles within specific narrow ranges may underscore structural features that could be correlated with specific characteristics of the datasets.

To delve deeper into the analysis of these patterns and study the observed ranges more comprehensively, a normalized loop count metric was introduced. This metric normalizes the number of cycles within each connected component based on the number of edges in that component:

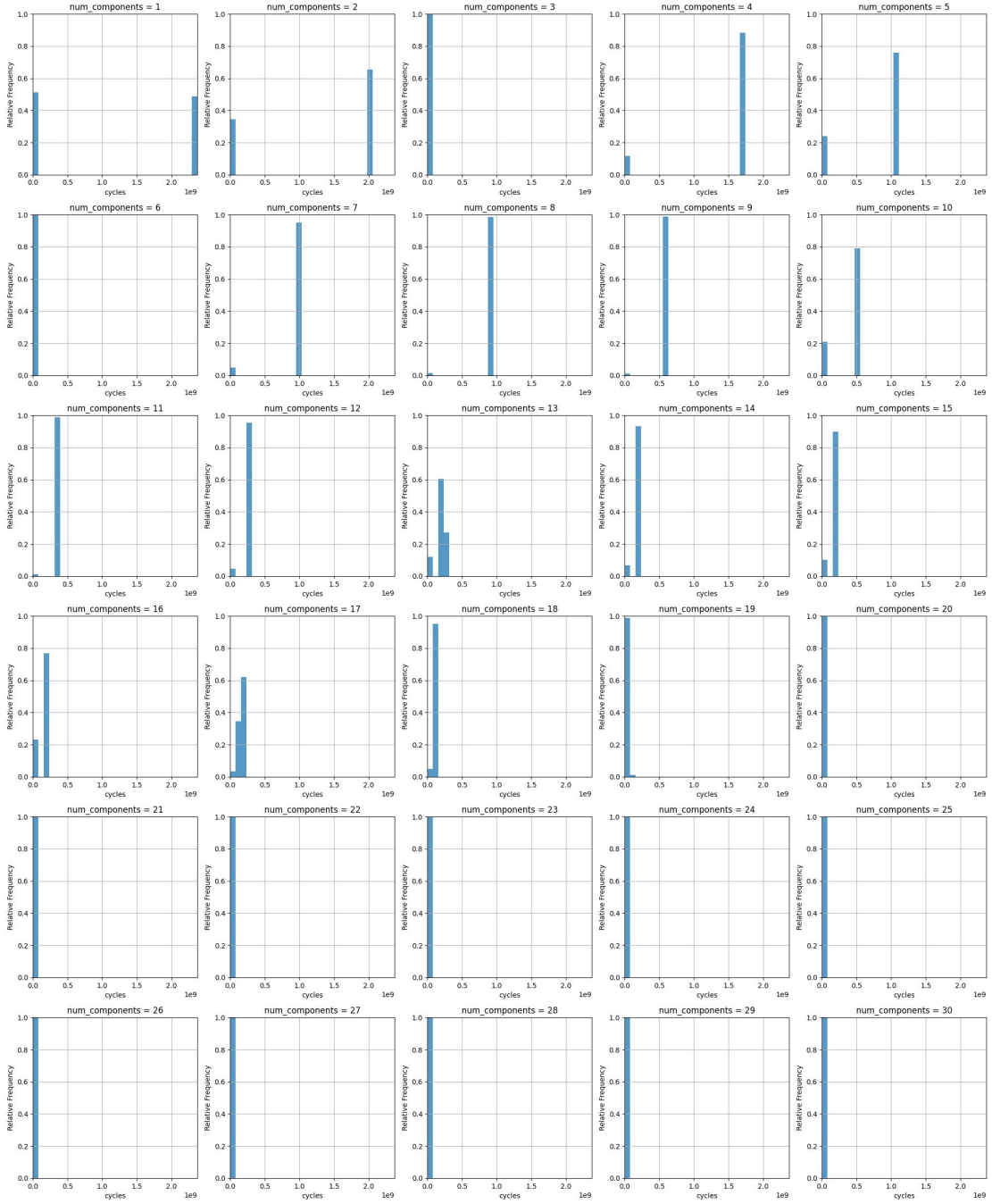

Figure 5.9: Histograms of the original number of loops for group 5. Each subplot corresponds to a specific number of connected components (num_components). The x-axis represents the number of cycles, and the y-axis shows the relative frequency.

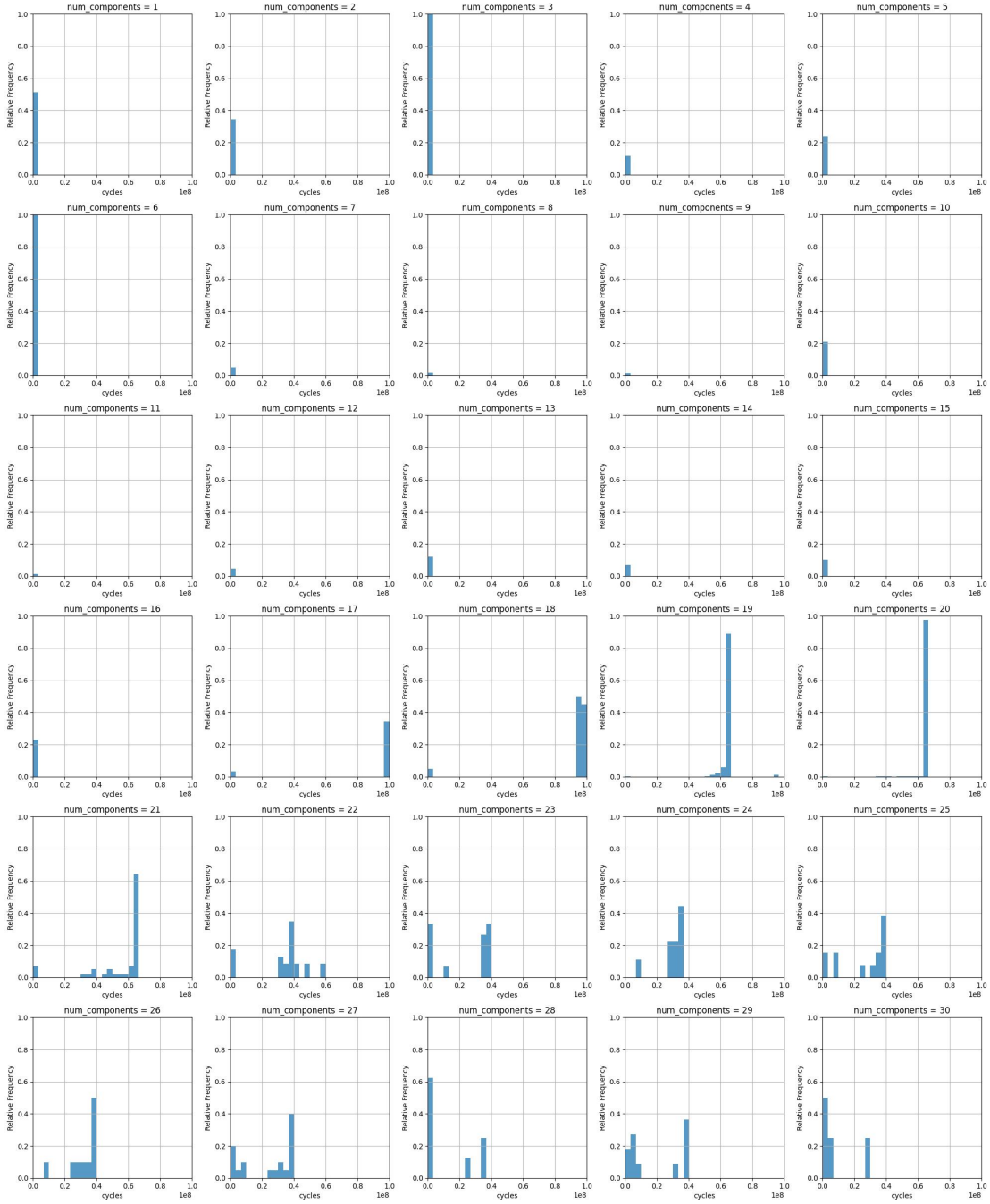

Figure 5.10: Histograms of the original number of loops for group 5 with range up to $1 \cdot 10^8$. Each subplot corresponds to a specific number of connected components (num_components). The x-axis represents the number of cycles, and the y-axis shows the relative frequency.

$$Loops_{norm} = \sum_{i \in C} \frac{Number\ of\ Loops_i}{Number\ of\ Edges_i}, \tag{7}$$

where C is a set of connected components in the graph, $Number\ of\ Loops_i$ is the number of loops in the i-th connected component, $Number\ of\ Edges_i$ is the number of edges in the i-th connected component. The normalized loop count provides information on the distribution of structural complexity across connected components, ensuring that no single component dominates disproportionally on the graph.

Studying this metric across the data groups discovers that for certain values of the connected component, the metric consistently reaches its maximum value, which means a high balance of the ε-graph. This parameter provides insights into the potential number of clusters in the data. For instance, in group 3, the normalized number of cycles frequently reaches its maximum value for connected component counts of 8, 9, 10, and 12 (Figure 5.11). Combining this information with a histogram of the frequency of the number of connected components (Figure 5.4), it can be concluded that the most likely number of clusters in this dataset is 12 as it has the highest peak out of these numbers.

The study of topological features of persistent homology, such as the number of cycles and its transformation, the number of cycles normalized by the number of edges per component, provides more precise insight into the actual number of clusters in the dataset. Knowing the number of connected components, it is possible to determine an optimal combination of DBSCAN parameters, such as minPts and ε.

5.3.3 The Similarity Metrics

By this stage of the experiments, it is possible to make a hypothesis about the appropriate number of connected components in the dataset. Knowing this value allows the extraction of the corresponding pairs of DBSCAN parameters minPts and ε. This part introduces an additional method to validate the results. Evaluating the quality of the clustering outcomes is crucial to ensure that the identified clusters accurately reflect the underlying structure of the data. To achieve this, the following widely used metrics are used, providing a comprehensive assessment of clustering quality:

- **Normalized Mutual Information (NMI)** is a statistical measure used to evaluate the similarity between two data clusterings. It quantifies the amount of shared information between the cluster assignments of data points in two clustering results, normalized to ensure that the measure is bounded between 0 and 1, where 1 indicates perfect alignment of the clusterings, and 0 indicates no shared information.

- **Adjusted Mutual Information (AMI)** is a statistical measure used to evaluate the similarity between two data clusterings, correcting for chance alignments. It calculates the mutual information shared by the cluster assignments, adjusted to ensure that the result is not inflated by random agreements, and is bounded between -1 and 1, where higher positive values denote stronger agreement.

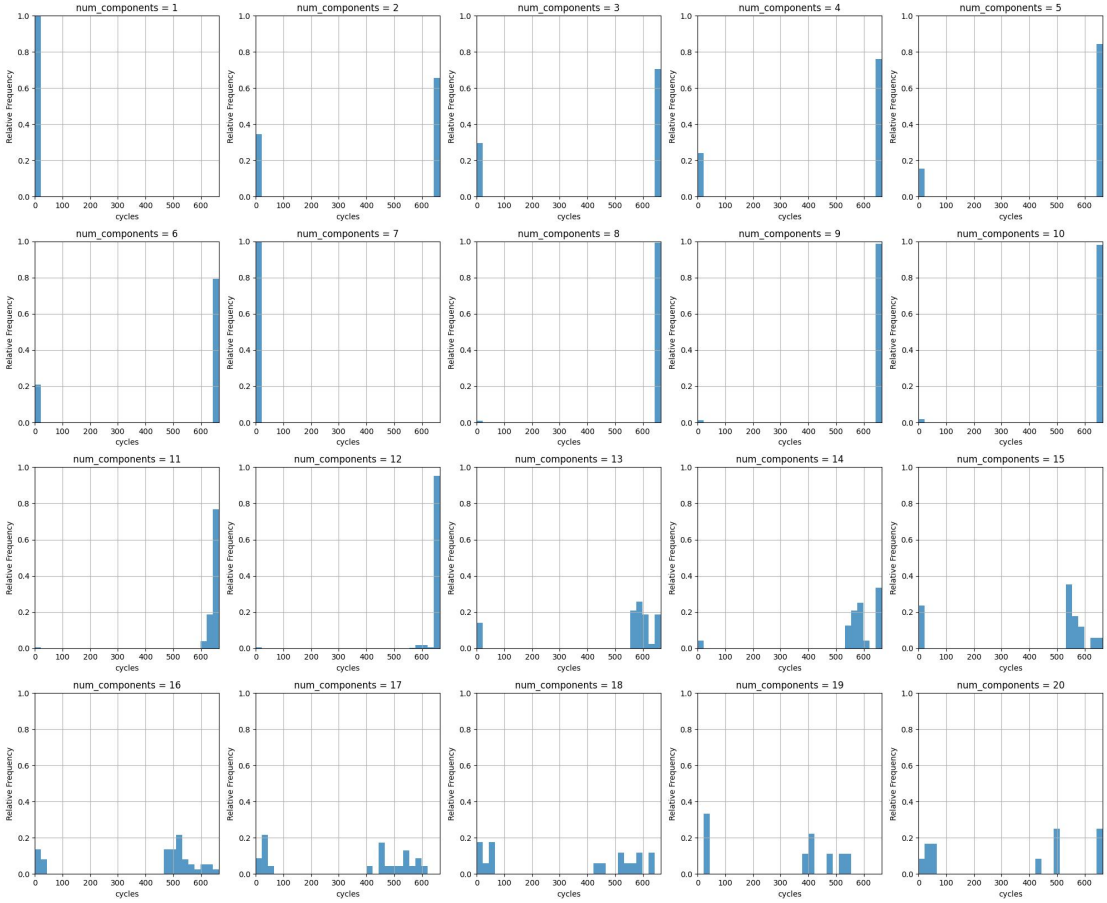

Figure 5.11: Histograms of the normalized number of loops for group 3. Each subplot corresponds to a specific number of connected components (num_components). The x-axis represents the number of cycles, and the y-axis shows the relative frequency.

- **Adjusted Rand Index (ARI)** is a statistical measure that evaluates the agreement between two data clusterings. It calculates the proportion of correct clustering decisions (pairs of points correctly placed in the same or different clusters) adjusted for chance, and is also bounded between -1 and 1, where higher values represent stronger similarity.

Comparing the connected components of one ε-graph with those of other ε-graphs that have the same number of connected components, we get an indicator of the similarity of cluster configurations. The closer this measure is to 1, the higher the likelihood that the clustering closely represents the true structure of the dataset, since there are no significant differences in clustering.

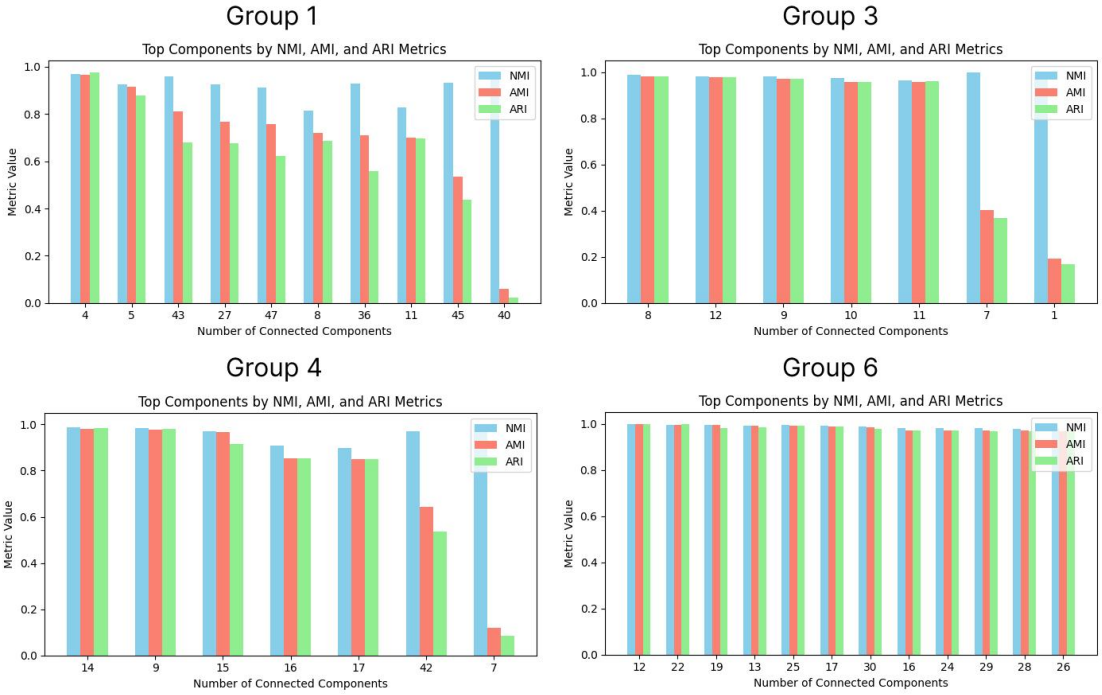

Figure 5.12: NMI, AMI, ARI plot for groups 1, 3, 4, and 6

For each dataset group, the NMI, AMI, ARI metrics were calculated for each connected component, and the top results are presented in Figures 5.12, 5.13 and 5.14. High values across all metrics indicate a reliable clustering structure, while difference suggest variability in the clustering results.

- **Group 1**: The highest similarity is observed for configurations with 4 and 5 connected components.
- **Group 2**: The highest similarity is observed for configurations with 10, 9 and 3 connected components.
- **Group 3**: The highest similarity is observed for configurations with 8 - 12 connected components.
- **Group 4**: The highest similarity is observed for configurations with 14, 9 and 15 connected components.
- **Group 5**: The highest similarity is observed for configurations with 20, 19, 8, 9 and 11 connected components.
- **Group 6**: The highest similarity is observed for configurations with 12-13, 16-17, 19, 22, 24-26, 28-30 connected components.

These results further confirm the conclusions drawn in Sections 5.3.1 and 5.3.2 for each of these groups. The comprehensive analysis of the topological features of persistent homology for specific parameter intervals provides insights into the data structure and helps identify an appropriate number of clusters. With this information, it becomes possible to determine the optimal combinations of DBSCAN parameters.

Moreover, this method has been proven to be robust. Experiments were conducted on noise-containing datasets, and even in these cases, the highest similarity values corresponded to the same number of connected components. For example, in noisy group 2, the highest similarity value is observed for 9 and 10 connected components (Figure 5.13). Similarly, in the noisy group 5, 19 and 20 connected components retain a high level of similarity between the clustering results (Figure 5.14).

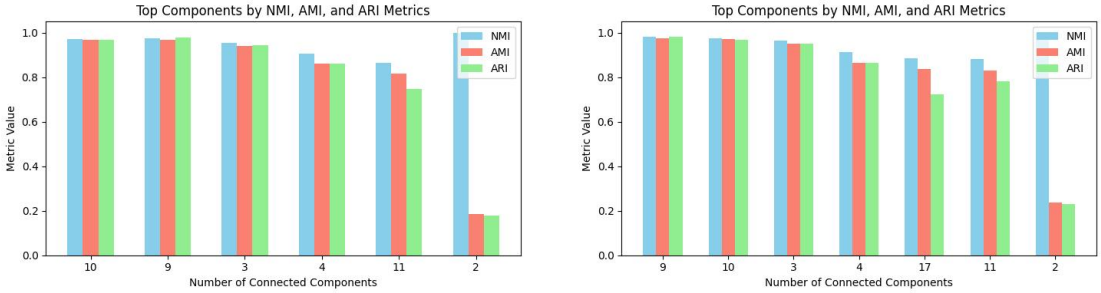

Figure 5.13: NMI, AMI, ARI plot for group 2. Left picture illustrates the plot for clean data, the right – the noisy data (1%).

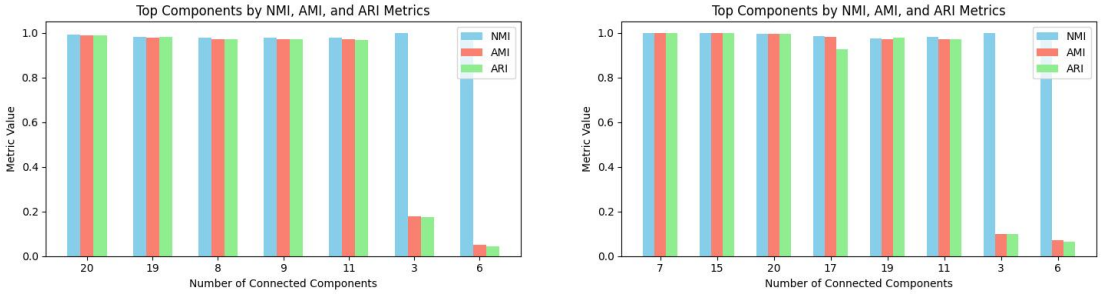

Figure 5.14: NMI, AMI, ARI plot for group 5.Left picture illustrates the plot for clean data, the right – the noisy data (3%).

5.4 Define the Range of minPts and ε

In these experiments, the focus is to investigate the question from Section 4.3 how determining the parameters minPts and ε affects both the stability and accuracy of the topological features discussed in Section 5.3. The choice of these parameters is critical in constructing the graph representation of the data and, consequently, the extracted topological features. A graph was constructed for each dataset where edges between nodes (data points) based on their dc-distance (Definition 11). An edge was created between two nodes if their dc-distance was less than a predefined ε value. The parameter minPts limits the graph's structure, determining the minimum number of connections that each node must have to be considered as a part of a cluster.

The choice of both minPts and ε parameters plays a key role in the resulting graph structure and consequently in the analysis of important topological features, such as connected components and loops. To explore the effect of these parameters a series of experiments were conducted using the datasets from groups 1 and 4, varying both minPts and ε across different settings to observe their impact. For minPts, three intervals were evaluated: $[2, 20]$, $[2, 30]$, $[2, 50]$. For ε, three ranges were tested: all possible dc-distances and the average distance between the 5th and 10th nearest neighbors. This experimental design resulted in nine distinct setups for each dataset, covering all combinations of minPts and ε values that are summarized in Table 5.2. Through these experiments, it can be determined that particular settings of minPts and ε can lead to more stable and accurate topological features across different datasets, potentially guiding future work in this area.

Initial Ranges				**DBSCAN parameters**	
minpts		ε		**minPts**	ε
				$[2, 20]$	all dc-dist
				$[2, 20]$	5 pt avg(all dc-dist)
				$[2, 20]$	10 pt avg(all dc-dist)
$[2, 20]$		all dc-dist		$[2, 30]$	all dc-dist
$[2, 30]$	x	5 pt avg(all dc-dist)	=	$[2, 30]$	5 pt avg(all dc-dist)
$[2, 50]$		10 pt avg(all dc-dist)		$[2, 30]$	10 pt avg(all dc-dist)
				$[2, 50]$	all dc-dist
				$[2, 50]$	5 pt avg(all dc-dist)
				$[2, 50]$	10 pt avg(all dc-dist)

Table 5.2: Parameter configurations for evaluating the impact of minPts and ε on stability and accuracy of connected components

The results for group 1 are presented in Figure 5.15. The observed trends are consistent regardless of the parameter variations. The top 3 most frequently occurring numbers of connected components are 4, 1, and 5, respectively. The peak at 4 indicates that four clusters are reliably detected, while the potential fifth cluster often appears as noise because it contains too few points to be recognized as a separate cluster (purple cluster

in group 1 in Figure 5.1). The secondary frequent value 1 includes cases where all points merged into a single cluster, which typically occurred for larger parameter values. As the minPts and ε values increase, the method becomes less sensitive to the local density of the clusters. The true number of connected components, 5, appears as the third most frequent value across experiments, demonstrating the robustness of the algorithm in identifying the correct number of clusters. These results indicate that the frequency of the number of connected components stays stable across a wide range of parameter settings.

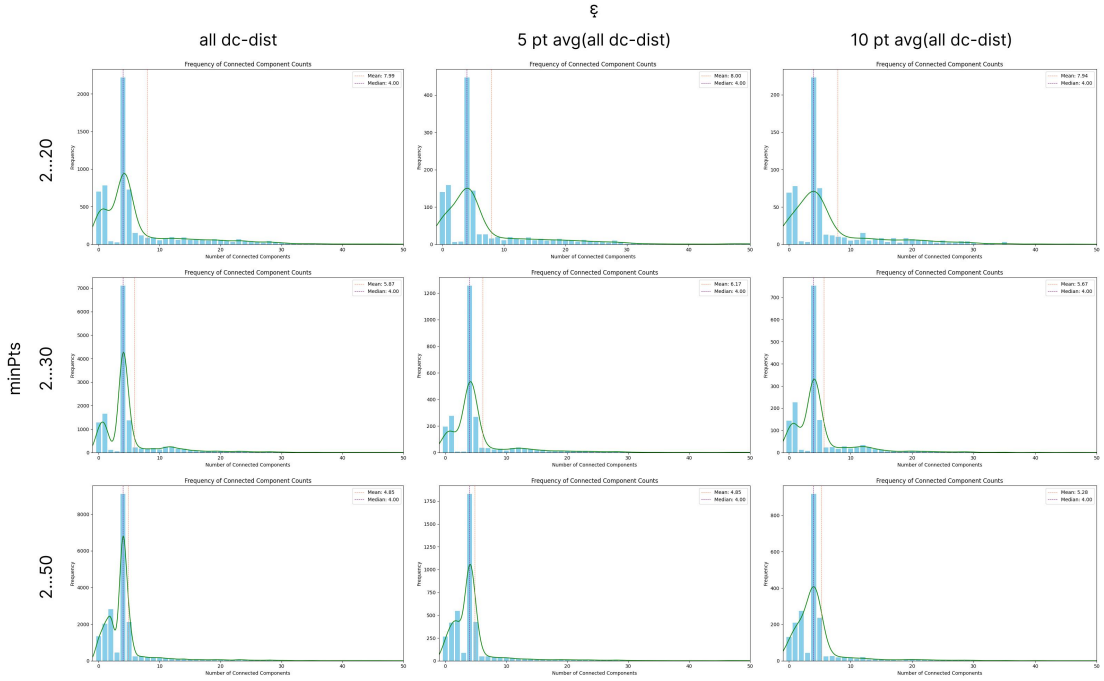

Figure 5.15: Histograms of the frequency of number of connected components for group 1 that present different combination of ε and minPts ranges.

By analyzing the experimental results for group 4 across different combinations of minPts and ε ranges from Table 5.2, a similar trend appears in the histograms in Figure 5.16. The general pattern remains stable across different ε values, however, there are slight changes with varying minPts ranges. Consider the ε as all dc-distances array and minPts as [2, 20] range there are four clear peaks at 14, 15, 9, and 0. These peaks stay stable within the minPts range changing, however, the frequency of the number of connected components between peaks increases with the extension of minPts range. For minPts [2, 50] the algorithm detects 6-13 clusters proportionally more than for minPts [2, 30] and [2, 20] ranges. Because of higher minPts values, the closely spaced smaller clusters are merged into larger ones, consequently, frequencies for values like 6, 11, and 12 become more visible. Despite this minor fluctuation, the correct number of connected components remains among the top-ranked frequencies, indicating the robustness of the clustering results. The top 3 most frequently occurring numbers of connected components are 14,

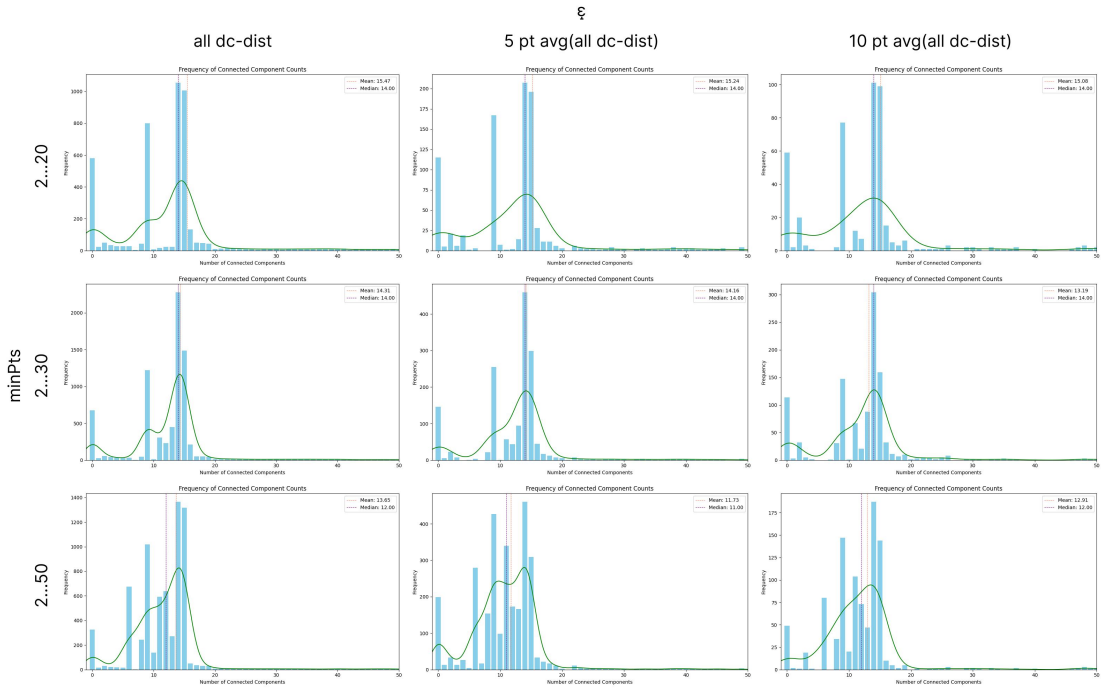

Figure 5.16: Histograms of the frequency of number of connected components for group 4 that present different combination of ε and minPts ranges.

15, and 9, respectively. The peaks at 14 and 15 mostly have close frequencies. It indicates that these values may represent the possible number of clusters in these datasets.

The analysis of these experiments highlights the significant role of the parameters minPts and ε in graph construction and the stability and accuracy of the topological features extracted from the dataset ε-graph. For both group 1 and group 4, the results demonstrate consistent trends: the clustering algorithm maintains robustness across a variety of parameter settings while identifying the true number of clusters in the data with high frequency. In addition, the observations show how higher minPts values tend to combine smaller clusters into larger ones. Consequently, this process slightly changes the distribution of detected connected components, but still preserves the true number of clusters among the most frequent outcomes.

6 Conclusion

This work explored the integration of persistent homology with density-based clustering algorithms, focusing on improving parameter selection. Through the analysis of synthetic datasets, the research demonstrates how persistent homology can provide valuable insights that can help to define optimal parameters for DBSCAN. The results highlightes the ability of topological features such as connected components and loops to improve the understanding of clustering outcomes and their stability across various settings. Moreover, the experiments demonstrated that persistent homology could offer a robust framework for analyzing clustering stability in the presence of noise and outliers.

Overall, this work provides a foundation for further research in the field of clustering, especially in density-based algorithms. The findings also underline the potential of combining topological data analysis with clustering techniques to solve current challenges.

6.1 Future Work

This research demonstrates the potential of integrating persistent homology with density-based clustering algorithms such as DBSCAN to improve parameter selection and improve clustering robustness. However, there are several directions for future work.

One of the directions for future research is to implement these analysis in large, high-dimensional datasets to evaluate scalability and performance in more complex scenarios. Additionally, applying the proposed framework to real-world datasets in different domains would help validate its practical utility and effectiveness. Further investigations could also focus on a more detailed analysis of the robustness to noise and outliers, examining how these factors affect the stability of topological features.

Another important direction involves analyzing higher-dimensional topological features, such as voids. These features may capture more complex cluster structures and provide deeper insights into the data topology, extending beyond connected components and loops to uncover more complex patterns in the data.

Finally, further work could investigate optimal strategies to initialize key DBSCAN parameters ε and minPts. This includes exploring effective ways to define parameter ranges, understanding their limitations, and developing systematic methods for parameter tuning to ensure robust, consistent, and valuable analyses as is described in this work.

By addressing these areas, future research can enhance both the practical applicability and theoretical foundations of density-based clustering methods enhanced with topological data analysis.

Bibliography

[1] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[2] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017.

[3] Gbeminiyi John Oyewole and George Alex Thopil. Data clustering: application and trends. *Artificial Intelligence Review*, 56(7):6439–6475, 2023.

[4] Daniel Brown, Arialdis Japa, and Yong Shi. An attempt at improving density-based clustering algorithms. In *proceedings of the 2019 ACM Southeast conference*, pages 172–175, 2019.

[5] Ricardo JGB Campello, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2):e1343, 2020.

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96 (34), pages 226–231, 1996.

[7] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

[8] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[9] Abir Smiti and Zied Elouedi. Dbscan-gm: An improved clustering method based on gaussian means and dbscan techniques. In *2012 IEEE 16th international conference on intelligent engineering systems (INES)*, pages 573–578. IEEE, 2012.

[10] Artur Starczewski and Andrzej Cader. Determining the eps parameter of the dbscan algorithm. In *Artificial Intelligence and Soft Computing: 18th International Conference, ICAISC 2019, Zakopane, Poland, June 16–20, 2019, Proceedings, Part II 18*, pages 420–430. Springer, 2019.

[11] Ulderico Fugacci, Sara Scaramuccia, Federico Iuricich, Leila De Floriani, et al. Persistent homology: a step-by-step introduction for newcomers. In *STAG*, pages 1–10, 2016.

[12] Yueqi Cao, Prudence Leung, and Anthea Monod. k-means clustering for persistent homology. *Advances in Data Analysis and Classification*, pages 1–25, 2024.

[13] Omer Bobrowski and Primoz Skraba. Cluster persistence for weighted graphs. *Entropy*, 25(12):1587, 2023.

[14] Peter Macgregor and He Sun. A tighter analysis of spectral clustering, and beyond. In *International Conference on Machine Learning*, pages 14717–14742. PMLR, 2022.

[15] Claudia Malzer and Marcus Baum. A hybrid approach to hierarchical density-based cluster selection. In *2020 IEEE international conference on multisensor fusion and integration for intelligent systems (MFI)*, pages 223–228. IEEE, 2020.

[16] Ahmed Fahim. Homogeneous densities clustering algorithm. *IJ Information Technology and Computer Science*, 10(10):1–10, 2018.

[17] Rong Guo and Jinsong Zhang. Research on 5g communication station location planning and regional clustering based on k-medoids and dbscan algorithm. In *2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*, pages 1097–1101. IEEE, 2022.

[18] Zhou Zichen, Luo Liangxiao, Li Zhenlin, Chao Miaomiao, Tang Wenchu, and Wang Yang. Harmonic pollution zoning method based on improved dbscan clustering. In *2022 China International Conference on Electricity Distribution (CICED)*, pages 1099–1103. IEEE, 2022.

[19] Nadia Rahmah and Imas Sukaesih Sitanggang. Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra. In *IOP conference series: earth and environmental science*, volume 31 (1), page 012012. IoP Publishing, 2016.

[20] Anna Beer, Andrew Draganov, Ellen Hohma, Philipp Jahn, Christian MM Frey, and Ira Assent. Connecting the dots–density-connectivity distance unifies dbscan, k-center and spectral clustering. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 80–92, 2023.

[21] Geoffrey Stewart and Mahmood Al-Khassaweneh. An implementation of the hdbscan* clustering algorithm. *Applied Sciences*, 12(5):2405, 2022.

[22] Yewang Chen. Dbscan is semi-spectral clustering. In *2020 6th International Conference on Big Data and Information Analytics (BigDIA)*, pages 257–264, 2020.

[23] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[24] Mehdi Karimi and Amir H Banihashemi. Message-passing algorithms for counting short cycles in a graph. *IEEE Transactions on Communications*, 61(2):485–495, 2012.

[25] Nur Fariha Syaqina Zulkepli, Mohd Salmi Md Noorani, Fatimah Abdul Razak, Munira Ismail, and Mohd Almie Alias. Topological characterization of haze episodes using persistent homology. *Aerosol and Air Quality Research*, 19(7):1614–1624, 2019.

[26] Mário Antunes, Joana Ribeiro, Diogo Gomes, and Rui L Aguiar. Knee/elbow point estimation through thresholding. In *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 413–419. IEEE, 2018.

[27] Benjamin Ertl, Jörg Meyer, Matthias Schneider, and Achim Streit. Coexdbscan: Density-based clustering with constrained expansion. In *KDIR*, pages 104–115, 2020.

[28] Jiawei Han, Jian Pei, and Hanghang Tong. *Data mining: concepts and techniques*. Morgan kaufmann, 2022.

[29] Alexander Rolle and Luis Scoccola. Stable and consistent density-based clustering via multiparameter persistence. *arXiv preprint arXiv:2005.09048*, 2020.

[30] Ashley Suh, Mustafa Hajij, Bei Wang, Carlos Scheidegger, and Paul Rosen. Persistent homology guided force-directed graph layouts. *IEEE transactions on visualization and computer graphics*, 26(1):697–707, 2019.

[31] Philipp Jahn, Christian MM Frey, Anna Beer, Collin Leiber, and Thomas Seidl. Data with density-based clusters: A generator for systematic evaluation of clustering algorithms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 3–21. Springer, 2024.