

JOINT INSTITUTE FOR NUCLEAR RESEARCH

Dzelepov Laboratory for Nuclear Research

**FINAL REPORT ON THE**

**START PROGRAMME**

*Implementation of DBSCAN algorithm in SPDRoot*

**Supervisor:**

Dr. Alexander Verkheev

**Student:**

Ilia Polishchuk, Russia  
Tomsk State University

**Participation period:**

February 11 –March 23,

Winter Session 2024

Dubna, 2024

Abstract

This study investigates the application of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) in high-energy physics experiments conducted at the SPD NICA facility. Using Monte Carlo-generated datasets simulating diverse particle interactions, DBSCAN's efficacy in clustering particle tracks and identifying decay chains is examined. Through empirical analyses and graphical representations, DBSCAN's performance is evaluated under various parameter configurations. The study highlights DBSCAN's ability to discern particle trajectories and detect decay products within complex event data. Results underscore the importance of parameter selection in optimizing clustering outcomes and suggest DBSCAN as a valuable tool for data analysis in high-energy physics experiments. Future research directions include exploring DBSCAN's compatibility with other clustering methods and modifications of the algorithm to mitigate its limitations.

Table of Contents

1. Introduction……………………………………………………………………………...4
2. DBSCAN…………………………………………………………………………………….5
   1. General Information…………………………………………………………..5
   2. Algorithm……………………………………………………………………………6
   3. Parameter Estimation…………………………………………………………7
   4. Advantages and Disadvantages…………………………………………..8
3. Results……………………………………………………………………………………..10
4. Conclusions……………………………………………………………………………...13
5. Acknowledgments…………………………………………………………………….14
6. Sources……………………………………………………………………………………..15

Introduction

Clustering algorithms serve as indispensable tools in data analysis, particularly for identifying inherent structures and patterns within datasets. Among these algorithms, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) stands out for its ability to cluster points based on their density relationships rather than predefined geometric shapes. In this report, we delve into the principles and applications of DBSCAN, focusing on its operational mechanisms, parameterization, advantages, and limitations.

DBSCAN operates by partitioning the data into clusters of varying densities, enabling the identification of core particle tracks, transient associations, and noise signals within the detector's output. By leveraging the density-based approach, DBSCAN effectively discerns meaningful clusters from background noise and facilitates the reconstruction of particle trajectories and decay chains, crucial for characterizing the underlying physics processes.

However, the successful utilization of DBSCAN hinges on accurate parameter estimation, including the selection of the neighborhood radius (ε) and the minimum points required for core point designation (m). Optimizing these parameters is essential for achieving optimal clustering performance and ensuring the reliable reconstruction of particle tracks and interactions.

Despite its robustness to noise and flexibility in handling arbitrary cluster shapes, DBSCAN has its limitations, including challenges with varying density datasets, computational intensity, and sensitivity to parameter selection. Careful consideration of these factors is necessary to harness DBSCAN's strengths effectively and mitigate its limitations in practical applications.

In this report, we explore the principles, applications, and considerations associated with the implementation of DBSCAN in the context of particle physics experiments conducted at the SPD NICA facility. Through empirical evaluations and case studies, we aim to provide insights into the capabilities and limitations of DBSCAN and offer guidance on its application in SPD NICA experiment.

* 1. General Information

Consider a set of points in some space to be clustered. Let ε be a parameter that defines the radius of a neighborhood surrounding some point. In context of DBSCAN clustering, these points are classified as core points, (directly-) reachable points and outliers, as follows:

* A point *p* is a *core point* if at least m (minimal amount of neighboring points) points are within distance *ε* of it (including *p*).
* A point *q* is considered *directly reachable* from *p* if point *q* is within distance *ε* from core point *p*. It is important to note that points are only considered directly reachable from core points.
* A point *q* is *reachable* from *p* if there is a path *p*1, ..., *pn* with *p*1 = *p* and *pn* = *q*, where each *pi*+1 is directly reachable from *pi*. Note that this implies that the initial point and all points on the path must be core points, with the possible exception of *q*.
* Any points that cannot be reached from another point are deemed outliers or noise points.

In the scenario where p represents a core point, it forms a cluster comprising all points reachable from it, regardless of whether they are core or non-core points. While every cluster must contain at least one core point, non-core points can also be part of a cluster, albeit they typically constitute its periphery, as they lack the capacity to reach additional points.

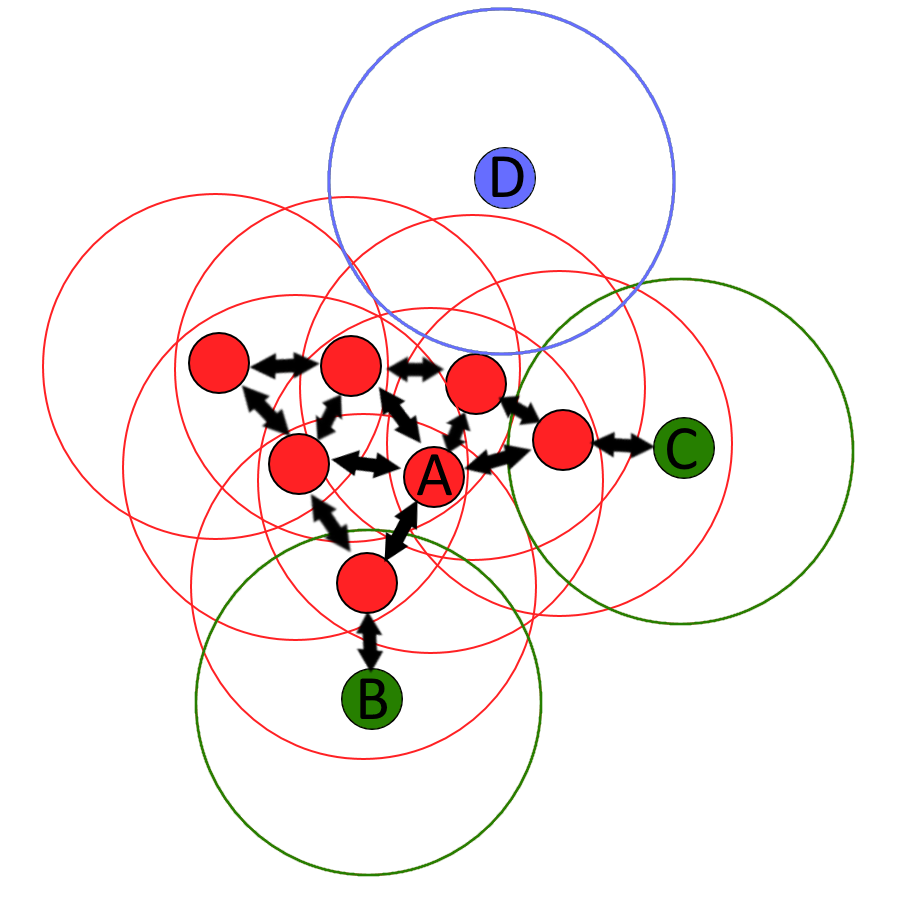


Figure 1. Illustration of DBSCAN cluster analysis

For instance, in Figure 1, where m = 4, point A and other red points are core points due to the presence of at least 4 neighboring points within the ε radius, forming a cohesive cluster since they are mutually reachable. Points B and C, although not core points, are reachable from A through other core points, hence included in the same cluster. On the other hand, point D, classified as a noise point, neither qualifies as a core point nor directly reachable.

It's important to acknowledge that the notion of reachability is asymmetric: only core points can reach non-core points, while the converse isn't true. Thus, a supplementary concept of density-connectedness is introduced to precisely define the scope of clusters identified by DBSCAN. Two points, p and q, are deemed density-connected if there exists another point o from which both p and q are reachable. This property of density-connectedness holds symmetrically.

* 1. Algorithm

DBSCAN operation necessitates two key parameters: ε and the minimum number of points essential to establish a dense region. Initially, an arbitrary unvisited point is chosen, and its ε-neighborhood is examined. If this neighborhood contains a sufficient number of points, a cluster is initiated; otherwise, the point is designated as noise. It's noteworthy that this point may later become part of a cluster if it's found within the ε-neighborhood of another sufficiently dense point.

Upon identifying a dense portion of a cluster, all points within its ε-neighborhood are incorporated into the cluster, along with their respective ε-neighborhoods if they also meet the density criterion. This iterative process continues until the entire density-connected cluster is identified. Subsequently, another unvisited point is selected for exploration, potentially leading to the discovery of additional clusters or noise.

DBSCAN's flexibility extends to its compatibility with various distance functions, similarity metrics, or other predicates, allowing practitioners to tailor the algorithm to suit diverse datasets and analytical requirements.

DBSCAN operates by iteratively exploring the dataset:

1. It starts with an arbitrary unvisited point and retrieves its ε-neighborhood.
2. If the neighborhood contains at least m points, a new cluster is initiated.
3. All points within this neighborhood are added to the cluster, along with their respective ε-neighborhoods if they are also dense.
4. The process continues until the density-connected cluster is fully explored.
5. Then, a new unvisited point is chosen, and the process repeats until all points are visited.
   1. Parameter Estimation.

Every data mining task encounters the challenge of parameterization, with each parameter exerting specific effects on algorithmic behavior. In the context of DBSCAN, the parameters ε and m hold paramount importance. These parameters necessitate user specification, with ε typically dictated by the problem domain, such as a physical distance measure, while m signifies the desired minimum cluster size.

Regarding m, a heuristic guideline suggests establishing a minimum value derived from the dataset's dimensionality D, with m ≥ D + 1. Assigning m to 1 is inherently illogical, as it would result in every point qualifying as a core point by definition. Similarly, m ≤ 2 leads to outcomes analogous to hierarchical clustering with the single link metric, truncated at ε height within the dendrogram. Thus, m must be set to at least 3. Nonetheless, larger values are often preferable for datasets afflicted with noise, as they tend to yield more robust clusters. A heuristic of m = 2·D may suffice, although larger values may be necessitated for expansive datasets, noisy data, or datasets containing numerous duplicates.

Determining the optimal value of ε entails plotting a k-distance graph, where the distance to the k = m-1 nearest neighbor is arranged from largest to smallest. Optimal ε values are discerned where the plot exhibits an "elbow": excessively small ε values lead to unclustered data segments, while excessively large ε values result in cluster merging, consolidating a majority of objects into a single cluster. Typically, smaller ε values are preferable, with only a fraction of points positioned within this distance of each other.

The selection of the distance function is intrinsically linked to ε determination and substantially influences the outcomes. Primarily, identifying a suitable measure of similarity for the dataset precedes ε determination. While no estimation exists for this parameter, the choice of distance function must be made judiciously in accordance with the dataset's characteristics.

* 1. Advantages and Disadvantages.

Advantages:

1. Robustness to Noise: DBSCAN demonstrates resilience to noise and outliers due to its density-based approach. By delineating clusters based on local density, it effectively disregards isolated data points that do not conform to the overall density pattern, thus enhancing the robustness of clustering outcomes.
2. Flexibility in Cluster Shape: Unlike partitioning methods such as K-means, DBSCAN is capable of identifying clusters of arbitrary shapes. This flexibility allows it to discern complex and irregularly shaped clusters, making it well-suited for datasets with non-linear structures or varying densities.
3. No Need for Prior Specification of Cluster Number: DBSCAN alleviates the need for a priori specification of the number of clusters within the dataset. By dynamically identifying clusters based on local density, it adapts to the inherent structure of the data, thus eliminating the requirement for user-defined cluster numbers.
4. Ability to Handle Outliers: DBSCAN explicitly identifies outliers or noise points within the dataset as objects that do not belong to any cluster. This feature is particularly advantageous in real-world datasets where noise and spurious data points are common, enabling more accurate and meaningful clustering results.

Disadvantages:

1. Sensitivity to Parameter Selection: DBSCAN's performance is highly dependent on the choice of parameters, particularly ε and m (minimum points). Selecting inappropriate parameter values can lead to suboptimal clustering results, necessitating careful parameter tuning, which may pose challenges, especially in high-dimensional or complex datasets.
2. Difficulty with Varying Density: DBSCAN may encounter challenges in datasets with varying densities, where clusters exhibit regions of differing densities. In such cases, selecting a single ε value that adequately captures the density variations across the dataset can be challenging, potentially leading to inaccuracies in cluster identification.
3. Computationally Intensive: DBSCAN's computational complexity can escalate significantly with the size and dimensionality of the dataset. As the algorithm involves calculating distances between data points and constructing neighborhoods, processing large-scale datasets may incur substantial computational costs, particularly when employing distance functions with high computational overhead.
4. Parameter Sensitivity to Data Scaling: DBSCAN's sensitivity to parameter selection is exacerbated by variations in data scaling. In datasets where features have vastly different scales, determining appropriate ε and m values becomes more intricate, potentially affecting the quality of clustering outcomes.

Another significant limitation associated with the SPD entails the necessity to perform event clustering on an individual basis, notwithstanding the utilization of a singular ε value selected for the entirety of the dataset, thereby rendering the adequate determination of ε impracticable.

In summary, while DBSCAN offers several advantages such as robustness to noise, flexibility in cluster shape, and adaptability to varying cluster numbers, its performance can be hindered by sensitivity to parameter selection, challenges with varying density datasets, computational complexity, and sensitivity to data scaling. Careful consideration and optimization of parameters are essential for leveraging the strengths of DBSCAN and mitigating its limitations in practical applications.

3. Results

During my participation in the JIRN START Programme, I undertook the task of DBSCAN clustering algorithm utilizing the C++ programming language. The utilization of a K-dimensional tree (K-d tree) facilitated efficient nearest neighbor search operations within the implemented algorithm.

The efficacy of my implementation was assessed through testing conducted on Monte Carlo-generated datasets, comprising diverse event types such as those involving muons, protons, and J/Ψ decays. Notably, the testing setup involved parameter configurations set to ε=15 and m=3 for comprehensive evaluation. Presented below are illustrative examples delineating the clustering outcomes observed within the analyzed events.

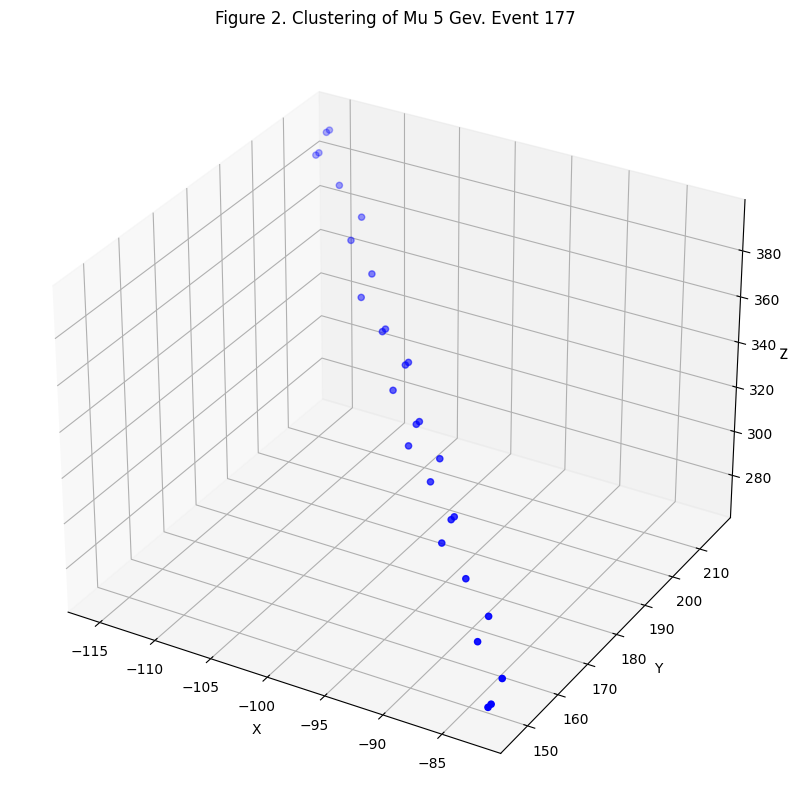


Figure 2 shows a graphical representation depicts the trajectory of a muon track, wherein all hits are consolidated within a singular cluster, aligning well with theoretical expectations.

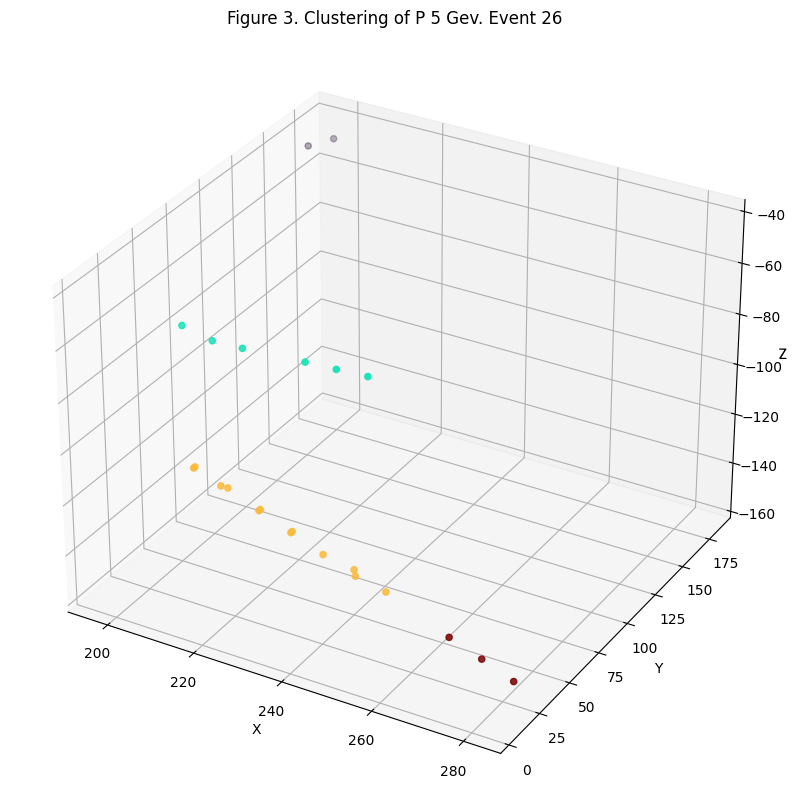


Figure 3 shows a scenario featuring two discernible tracks alongside two outlier observations. While the majority of the observations conform to theoretical predictions, a notable discrepancy arises wherein hits denoted in yellow and red fail to be associated with a single cluster. This discrepancy underscores a pivotal limitation of the algorithm, namely, its susceptibility to accurately capture density fluctuations within datasets. The selection of a singular ε value, which sufficiently encapsulates the dataset's density variations, presents a formidable challenge, thereby resulting in suboptimal cluster identification.

Furthermore, an increase in the ε value exacerbates the situation, leading to premature merging of cyan and yellow clusters prior to their convergence with the red cluster, thereby deviating from theoretical expectations.

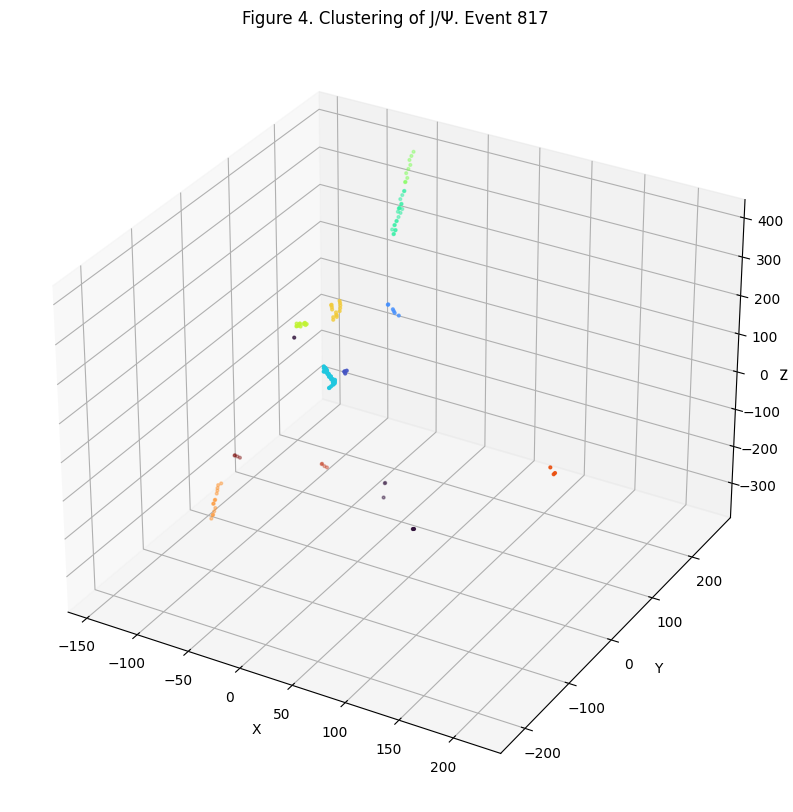


Figure 4 shows the decay products of J/Ψ into and . Analogous to previous instances, the observed clustering pattern deviates from theoretical predictions, wherein cyan and green hits fail to coalesce into a singular cluster due to the inherent limitation of DBSCAN.

To validate the hypothesis that inadequate cluster formation stems from inherent data gaps, an alternate implementation of DBSCAN from the scikit-learn (sklearn) library was employed on the same datasets. Notably, the outcomes yielded by this alternative implementation corroborated the findings obtained from the original algorithm, thus reaffirming the presence of inherent limitations within the DBSCAN methodology.

However, it is imperative to note that these limitations can be effectively mitigated through the adoption of Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN). HDBSCAN offers several advancements over DBSCAN, particularly in its ability to variate ε, thereby alleviating the need for its manual tuning.

The efficiency of the algorithm was also assessed in terms of operating time. The results are presented in a table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Number of events | Number of points | Time in seconds |
| μ 10GeV | 1000 | 20396 | 1,496 |
| J/Ψ 10GeV | 1000 | 71441 | 2,848 |
| p 5GeV | 10000 | 132860 | 4,322 |

As can be seen from the table, the operating time of the algorithm has a linear dependence on the number of points in the data.

Conclusion

In this study, we investigated the application of DBSCAN in high-energy physics experiments conducted at the SPD NICA facility. Through empirical analyses utilizing Monte Carlo-generated datasets simulating various particle interactions, we evaluated DBSCAN's efficacy in clustering particle tracks and identifying decay chains. Our findings underscored DBSCAN's ability to discern particle trajectories and detect decay products within complex event data, highlighting its potential as a valuable tool for data analysis in high-energy physics experiments.

While DBSCAN demonstrates robustness to noise and flexibility in handling arbitrary cluster shapes, we acknowledge its limitations, particularly its sensitivity to parameter selection, challenges with varying density datasets, and computational intensity. However, we propose that these limitations can be mitigated through the adoption of Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), an extension of DBSCAN designed to address some of its shortcomings.

In conclusion, our study contributes to the understanding of DBSCAN's capabilities and limitations in the context of particle physics research, providing insights that can inform the development of more effective clustering algorithms for data analysis in this domain.

Acknowledgments

Heartfelt thanks to my supervisor Alexander Verkheev: he always found time to answer my questions, even the most trivial ones. I am immensely grateful for his guidance, patience, and unwavering support throughout this journey. I have learned invaluable lessons from his vast expertise and dedication to nurturing my growth as a researcher.

Warm thanks go to JINR for providing me with this exceptional research opportunity and for their generous financial support. I am deeply appreciative of the resources and environment they have provided, which have been instrumental in the success of this endeavor.

Sources

1. Ester, Martin, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise." In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), pp. 226-231. AAAI Press, 1996. DOI: 10.5555/3001460.3001507.
2. Schubert, Erich, Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN." In ACM Transactions on Database Systems (TODS), vol. 42, no. 3, pp. 19:1-19:21. DOI: 10.1145/3068335.
3. Campello, Ricardo JGB, Davoud Moulavi, Arthur Zimek, and Jörg Sander. "Hierarchical density estimates for data clustering, visualization, and outlier detection." ACM Transactions on Knowledge Discovery from Data (TKDD) 10, no. 1 (2015): 1-51. DOI: 10.1145/2733381.
4. Hahsler, Michael, Kurt Hornik, and Christian Buchta. "Getting things in order: An introduction to the R package seriation." Journal of Statistical Software 25, no. 3 (2008): 1-34. DOI: 10.18637/jss.v025.i03.
5. Grira, Noura, and Adil Kabbaj. "DBSCAN revisited: performance analysis of the DBSCAN algorithm implementations." In 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), pp. 113-119. IEEE, 2014. DOI: 10.1109/AICCSA.2014.7073199.
6. Bentley, Jon Louis. "Multidimensional binary search trees used for associative searching." Communications of the ACM 18, no. 9 (1975): 509-517. DOI: 10.1145/361002.361007.