2023 International Conference on Automation, Control and Electronics Engineering (CACEE) | 979-8-3503-0920-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/CACEE61121.2023.00023

SS-DBSCAN: Epsilon Estimation with Stratified Sampling for Density-Based Spatial Clustering of Applications with Noise

Gloriana Joseph Monko
Department of Functional Control Systems
Graduate School, Shibaura Institute of Technology
Tokyo, Japan
nb22504@shibaura-it.ac.jp

Masaomi Kimura
Department of Computer Science and Engineering
Shibaura Institute of Technology
Tokyo, Japan
masaomi@shibaura-it.ac.jp

***Abstract*—Clustering algorithms are crucial in uncovering hidden patterns and structures within datasets. Among the density-based clustering algorithms, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) has gained considerable attention for its effectiveness in various applications. However, determining appropriate parameter values for this algorithm remains a challenging task. This paper presents a novel methodology for eps parameter estimation for an improved DBSCAN, namely SS-DBSCAN. The experimental results across nine datasets demonstrate the efficacy of our proposed method in accurately determining clusters with eps value from SS-DBSCAN algorithm. The clusters identified using estimated eps values by SS-DBSCAN align well with the inherent structure of the datasets, yielding better cluster results than the manually set parameters and other methods used for automatic estimations of the eps for DBSCAN. Our approach adapted well to the peculiarities of each dataset, whether dealing with different scales, dimensions, or densities; it proved the versatility and robustness across various datasets, thereby emphasizing its generalizability and potential for broader applications.**

***Index Terms*—SS-DBSCAN, eps estimation, density based clustering, stratified sampling, diverse dataset**

I. Introduction

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is an eminent unsupervised machine learning algorithm [1], widely acclaimed for its proficiency in a multitude of data-driven sectors including but not limited to computer vision, bioinformatics, speech processing, and anomaly detection [2]. DBSCAN [2] , presented at a data mining symposium, is a renowned density-centric clustering method. It has been developed multiple times independently and can be found in numerous clustering software packages, including scikit-learn [3], R [4], Weka [5], among others. Numerous density-centric techniques drew inspiration from DBSCAN, seeking to address its shortcomings [6], [7], [8], [9].

DBSCAN, as a density-based clustering algorithm, distinguishes itself from partition-based clustering methods like K-Means or hierarchical clustering methods. Its core strength lies in the capability to identify and create clusters of arbitrary shapes and sizes, which is a decisive factor when dealing with real-world data that does not always fall into neat spherical groups [10]. Beyond its ability to capture complex cluster structures, DBSCAN offers an additional advantage of robustly handling noise and outliers [11]. This characteristic greatly enhances the algorithim's effectiveness with real-world data sets that frequently have noise and outliers. The algorithm achieves this by categorizing data points into three groups: core, border, and noise points [12]. Fig. 1.

A. Problem

Despite these undeniable advantages, the DBSCAN algorithm has its share of challenges, the most notable of which revolves around selecting its key parameters: Epsilon (eps) and Minimum Points (MinPts). The optimal performance of DBSCAN is largely influenced by these parameters. The critical limitation, however, is that these parameters are not self-tuning. Instead, they require manual input, and the selected values directly and considerably impact the algorithm's output. A poor selection of these parameters can lead to a less accurate representation of data, resulting in suboptimal clusters, overfitting or underfitting, and a high misclassification rate of data points [13]. Choosing an appropriate eps value poses a particular challenge. A larger eps value might cause smaller clusters to merge, resulting in fewer larger clusters and potentially meaningful information getting lost. Conversely, a smaller eps value might lead to many data points being labeled as noise, thereby losing potential clusters. Similarly, the selection of MinPts is equally crucial. A greater MinPts value might result in more data points being identified as noise, whereas a smaller value might result in numerous, possibly less meaningful clusters [6]. Since selecting these parameters heavily relies on an in-depth understanding of the data and its inherent structure, finding the right balance becomes a formidable task. This challenge is further amplified by the fact that what works well for one dataset might not necessarily work for another, making a one-size-fits-all approach impractical [14]. This situation underscores the necessity for an automated, data-dependent approach for estimating these parameters. Such an approach would eliminate the need for

979-8-3503-0920-1/23/$31.00 ©2023 IEEE
DOI 10.1109/CACEE61121.2023.00023

manual tuning, enhance the applicability and accuracy of the DBSCAN algorithm, and result in more insightful clusters that truly represent the underlying data structure.

B. Contribution

This paper presents an improved DBCSAN algorithm namely SS-DBSCAN (Stratified Sampling for Density-Based Spatial Clustering of Applications with Noise) that utilizes stratified sampling for eps estimation. We use the stratified sampling technique to help determine the eps value by segmenting the data into homogeneous subgroups. Our method theorizes that the best way to find eps is to examine the distances between neighboring data points and target points within a representative subset of the original dataset. Given that the distances can vary depending on which points are randomly selected, we employ stratified sampling to ensure that our chosen subset accurately reflects the diversity of categories or groups within the overall dataset while mitigating the risk of selecting points only from densely populated regions and enables more reliable distance calculations between target data points and their neighbors. The experimented method aims to elevate the DBSCAN algorithm's potential to understand complex and varied cluster structures across different datasets without manual intervention.

Through this study, we aim to make a meaningful contribution to the field of density based clustering, especially in enhancing the practical application of the DBSCAN algorithm. By simplifying the parameter selection process, we aspire to make this robust clustering algorithm more user-friendly, adaptable, and essential for extracting meaningful insights from diverse datasets.

C. Key definitions and concepts

- Epsilon (eps) is the maximum allowable distance between two points for them to be categorized in the same cluster [15].
- Minimum samples (MinPts) indicate the least number of points needed within the neighborhood radius (eps) for a point to qualify as a core point in a dense area [15].
- Core points are those that have a minimum specified number of neighbors within an eps distance [15].
- An outlier or noise point is neither classified as a core nor a boundary point [15].
- Stratified sampling refers to statistical method that divides a population into subgroups or strata. In our context, the population refers to the dataset, and the subgroups or strata represent the inherent clusters within the dataset [16].
- In our study, we chose the ”knee point” method to find the best ”eps” value for our clustering algorithm. We opted for this method because it's simple, easy to understand, effective, and doesn't require much computational power. Other methods like cross-validation and grid search need more calculations since they run the algorithm many times with different settings. The knee point method helps us find an ”eps” value that groups as many data points as possible into clusters without adding too much noise or merging too many clusters [17].

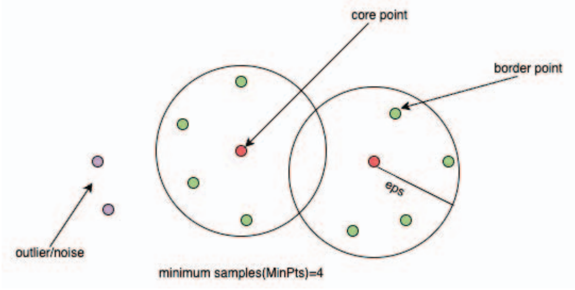

Fig. 1: DBSCAN Clusters

II. Related Works

Various studies have explored adaptations and enhancements of the epsilon estimation method, such as incorporating statistical techniques like Gaussian mixture models or employing data-dependent epsilon values based on local characteristics of the dataset.

Ester et al. [2], the pioneers of the DBSCAN algorithm, acknowledged the impact of selecting eps and MinPts on the clustering outcomes. However, their seminal work did not provide a concrete solution to automatically estimate these parameters, leaving the door open for future research. In a study by Sander et al. [10], the authors presented a method for determining eps by considering the ”k-distance” graph. This method measures the distance of each point to its k-th nearest neighbor and plot these distances in an increasing sequence. The ”elbow” or the point of maximum curvature in this plot proposes the optimal eps value. Though this method provides an intuitive means of choosing eps, it requires users to visually inspect and interpret the k-distance graph, which can be subjective.

Liu et al. [11] presented a modified version of DBSCAN called DBSCAN-DLP, in which they used a dynamic approach to select the eps value. They calculated the eps value for every data point by examining the point's local density and the mean distance of the point to all other points in the dataset. Despite improving the clustering performance, this approach increased the computational complexity due to the individual calculation of eps for every data point.

Karami and Johansson [18], put forth a novel enhancement called BDE-DBSCAN. This method ingeniously married the principles of Binary Differential Evolution with the original DBSCAN. The hybrid method they presented is lauded for its clever use of analytical strategies and the tournament selection method, aimed explicitly at fine-tuning DBSCAN's parameters, notably epsilon and minimum points. In a parallel vein of development, Ren et al. [19], sought to modify the underlying distance metric of DBSCAN. They introduced the DBCAMM variant which replaces the traditional Euclidean distance with the Mahalanobis distance. Further enriching this variation, they incorporated an innovative merging strategy, which proved to be especially adept at delivering enhanced visualization results for image segmentation tasks.

The optimization techniques applied to DBSCAN leaped with Lai et al. [20]. Their approach, underscored by the 'MVO-multiverse optimizer algorithm', was centered around iteratively refining DBSCAN parameters. This method offered an adaptive way of ensuring optimal clustering. Khan et al.'s [21] contribution was introducing an adaptive version of DBSCAN. Their method, aptly named adaptive DBSCAN, was designed meticulously to automate the process of determining the most suitable values for parameters such as epsilon and the number of minimum points.

Ram et al.'s [22] work emphasized a new density-varied algorithm that showed proficiency in addressing the local density variations present within clusters. Birant and Kut, in [23], unveiled the STDBSCAN algorithm. This technique, designed for spatial-temporal data, distinguished itself by its adeptness at detecting clusters and noise, especially in scenarios characterized by different data densities. Cheng et al. [24] introduced a grid-based approach for choosing the eps parameter. They divided the data space into grids and calculated the eps value based on the density of each grid. This approach was efficient but worked best with datasets that have uniform density, limiting its general applicability.

While these studies have significantly contributed to the field, they also highlight the need for a more adaptable, data-driven, and user-friendly approach to estimating eps values. The methodology proposed in this paper aims to address this need and further enhance the effectiveness and practical utility of the DBSCAN algorithm.

III. The SS-DBSCAN Algorithm

SS-DBSCAN employs a stratified sampling technique for the data by dividing it into several strata or subsets, where each stratum is internally homogenous and maximally different from other strata [25], [26]. After segmenting the data into various strata, we compute the mean distance between each pair of points inside each stratum and note down these average distances. The logic here is to ensure that all points within each stratum are reachable from one another and that different strata are in different clusters. Using stratified sampling for eps estimation helps mitigate the limitations of using a fixed eps value for the entire dataset.

The approach of estimating the epsilon (ϵ) value for a k-nearest distance graph using stratified sampling involves a five-step process:

Step 1: Let X be the sample data, where $X = \mathbb{R}^d$, and d represents the number of samples. Define μ as the distance between two elements of X. For this purpose, we utilize the Euclidean distance (1):

$$\mu = \sqrt{\sum_{i=1}^{d}(x_i - x_i')^2} \tag{1}$$

Step 2: Let $\pi_1(\mu) \dots \pi_k(\mu)$ represent the distances ordered in ascending order to their proximity to x, where k denotes the number of chosen nearest neighbors. We expressed this as:

$$\pi_1(\mu) \leq \pi_2(\mu) \leq \dots \leq \pi_k(\mu) \tag{2}$$

Step 3: Calculate the average distance (π_{avg}) for each element from its k neighbors using the following formula:

$$\pi_{avg}(\mu_d) = \frac{1}{k}\sum_{i=1}^{k}\pi_i(\mu_d) \tag{3}$$

The traditional k-distance plot uses the distance to the $k^{th}(\pi_k)$ neighbor of (2) to find the knee point (ϵ value). In contrast, our method employs the average distance (3) from a point to its k neighbors to determine the ϵ value. By focusing on this average distance, we capture a central tendency of the point's surroundings, offering a broader view of its local structure, especially in the presence of outliers. This distinction in our approach, provides a more holistic perspective on data relationships.

Step 4: Implement stratified sampling to obtain distances from each element's average distance from its nearest neighbors. Form strata (τ_p) (4) by subgrouping average distances into class intervals of equal size. Here, p represents the count of strata created, which aligns with the number of class intervals formed. This is expressed as:

$$\tau_p = \pi_{avg_{lower-limit}}(\mu_d) - \pi_{avg_{upper-limit}}(\mu_d) \tag{4}$$

Sample a specified number of distances (s) from each stratum using m elements sampled from each and r as sampling rate. This can be defined as (5) and (6):

$$s_p = \text{sample}(\tau_p, r) \tag{5}$$

$$\hat{s} = \{s_1, s_2, \dots, s_m\} \tag{6}$$

We use the calculated mean distance from each data point, X, to its k nearest neighbors and apply stratified sampling to ensure equal representation based on these distances. We introduce a function, F(x), which discretizes continuous data into specific bins or strata, aiding in clearer data visualization. For instance, it groups distances into ranges like 0-5, 6-11, and 12-17 by utilizing the concept of class intervals based on the specified number of bins. Each data point is then categorized into these bins. We then decide on a sampling rate, like 0.8, to randomly sample 80% of data from each stratum. If any sampling imbalances arise across bins, we balance it by up-sampling the underrepresented data. This ensures equal representation across strata, optimizing our algorithm's performance, especially for uneven datasets.

Step 5: Generate a k distance graph from the distances obtained through stratified sampling in (6). Using a knee locator technique, identify the epsilon (ϵ) value.

From Fig. 2 there is a difference that is notable in both DBSCAN and SS-DBSCAN plots: Using the k-th neighbor distance in DBSCAN creates a smoother curve due to consistent neighbor ordering. However, in our approach, SS-DBSCAN, the use of average distance results in a more jagged

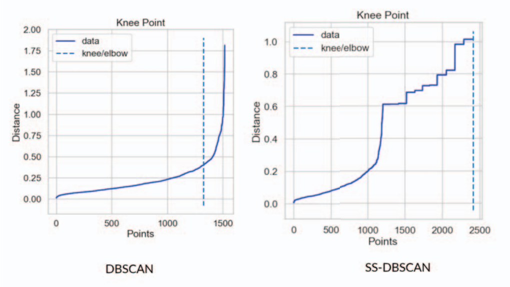

Fig. 2: K-distance graphs

curve because of neighborhood variations. This jaggedness in the average-distance plot captures detailed data structures more effectively. If a dataset contains clusters with different densities or shapes, the average distance is more attuned to these variations than the k-th neighbor distance. As a result, the average-distance method consistently provides superior eps values based on our evaluation metrics.

IV. Results from Experiments

We conducted experimental analyses using several benchmark datasets for clustering to demonstrate the efficacy of our method, SS-DBSCAN. These datasets were sourced from GitHub and included five artificial datasets (2d-20c-no0, elly_2d10c13s, sizes1, square4, and st900) and four real-world datasets (Iris, Sonar, Arrhythmia, and Iono). The datasets varied in size, dimensionality, and inherent cluster structures. Our main objective was to evaluate the resilience and flexibility of the method we introduced, which can be seen in the clustering results of each algorithm in Table I. By comparing the insights gathered from this comparison, we can shed light on the efficiency and accuracy of both clustering approaches.

A. Evaluation of the Artificial Datasets

- **2d-20c-no0 dataset:** SS-DBSCAN effectively identified 19 clusters, aligning with the dataset's 19 actual labels. In contrast, the conventional DBSCAN detected 35 clusters, demonstrating our method's superiority.
- **sizes1 dataset:** SS-DBSCAN accurately discerned 4 clusters, matching the 4 actual labels. In comparison, traditional DBSCAN could only identify a singular cluster.
- **square4 dataset:** The proposed SS-DBSCAN method consistently pinpointed 4 clusters, perfectly aligning with the 4 actual labels of the dataset. The traditional DBSCAN, however, identified only one cluster.
- **st900 dataset:** Here, SS-DBSCAN successfully detected 8 clusters, closely matching the dataset's actual label count of 9. Conversely, DBSCAN could only detect a single cluster.
- **elly_2d10c13s dataset:** Both algorithms identified just one cluster with outliers. This result underscores the notion that no single method can consistently outperform across all datasets.

B. Evaluation of Practical-world Datasets

- **Iris dataset:** While the dataset had 3 actual labels, both DBSCAN and SS-DBSCAN algorithms detected 2 clusters without any outliers.
- **Sonar dataset:** SS-DBSCAN detected 2 clusters, perfectly aligning with the true class labels of the dataset while DBSCAN returned 3 clusters.
- **Arrhythmia dataset:** SS-DBSCAN outperformed DBSCAN by detecting 12 clusters and outliers, closely mirroring the 13 actual labels. In contrast, DBSCAN only identified 2 clusters, accompanied by outliers.
- **Iono dataset:** SS-DBSCAN detected 3 clusters, whereas the actual dataset had 2 class labels. DBSCAN, on the other hand, could only recognize a single cluster.

This comprehensive analysis underscores the versatility and improved performance of our proposed SS-DBSCAN method compared to traditional DBSCAN.

V. Discussion

The central focus of this study was to introduce a novel methodology for optimizing parameter estimation, specifically, eps in the DBSCAN clustering algorithm. Our approach incorporates stratified sampling for eps estimation facilitated by a k-neighbors approach for eps estimations. According to the experimental results from nine diverse datasets, the proposed methodology substantially enhances the effectiveness of eps parameter estimation in DBSCAN clustering. Our findings strongly suggest that stratified sampling for eps estimation offers an edge over existing methods. The stratified sampling accounts for the inherent complexities of the dataset's structure and density variations, allowing for a more adaptive and region-specific determination of the eps parameter. This capability is substantiated by the consistently better cluster formation observed across all datasets, pointing to improved clustering outcomes. Additionally, the adaptability of this approach is highlighted by the flexibility it affords in parameter tuning (i.e. number of neighbors, k, and bin count, n). For our study, we determined the number of neighbors, k based on dataset size: larger datasets demanded a large k, and vice versa. As for the number of bins, although we recommend a default of 3, the methodology permits a range greater than three and not less than 2, thereby providing room for customization based on specific data characteristics.

Another notable advantage of our proposed methodology is its versatility and robustness across different datasets. Whether dealing with different scales, dimensions, or densities, our approach adapted well to the peculiarities of each dataset, thereby emphasizing its generalizability and potential for broader applications.

Lastly, DBSCAN and SS-DBSCAN have different clustering approaches, reflected in their eps values. While DBSCAN's k-th neighbor distance often leads to smaller eps values, SS-DBSCAN's use of average distances with stratified sampling results in larger eps values. It suggests SS-DBSCAN captures broader data structures, potentially forming larger or more inclusive clusters than DBSCAN.

TABLE I: Clustering Results for DBSCAN and SS-DBSCAN Against Actual Labels

	DBSCAN				SS-DBSCAN				
Dataset	**eps**	**MinPts**	**Outliers**	**Clusters**	**eps**	**MinPts**	**Outliers**	**Clusters**	**Actual lables**
2d-20c-no0	0.40150	4	173	35	1.01432	12	56	19	19
sizes	1.93697	4	3	1	2.78358	69	18	4	4
square4	1.57385	4	6	1	2.63267	122	86	4	4
st900	0.45121	4	1	1	0.68156	45	124	8	9
elly-2d10c13s	0.21982	4	1	1	0.53009	124	102	1	9
Iris	2.95133	4	0	2	2.16473	4	0	2	3
Sonar	2.17496	4	5	3	4.44850	31	42	2	2
Arrhythmia	3.02978	4	9	2	1.83828	4	58	12	13
lono	2.49644	4	1	1	2.72438	16	31	3	2

VI. Conclusion

Our study demonstrates that stratified sampling for eps significantly improves eps estimation in DBSCAN clustering algorithms. While our methodology has shown promising results, future research could explore the impact of other stratification techniques, examine scalability issues, or investigate how the methodology performs on large-scale datasets. By addressing the limitations inherent in traditional parameter selection methods for DBSCAN, this study contributes a robust and adaptable methodology that is easygoing to serve better the diverse challenges encountered in data clustering applications.

References

[1] W. Lai, M. Zhou, F. Hu, K. Bian, and Q. Song, "A New DBSCAN Parameters Determination Method Based on Improved MVO," IEEE Access, vol. 7, pp. 104085–104095, 2019, doi: 10.1109/ACCESS.2019.2931334.

[2] X. X. Martin Ester, Hans-Peter Kriegel, Jiirg Sander, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in KDD-96 Proceedings, 1996, pp. 226–231.

[3] M. Amiruzzaman, R. Rahman, M. R. Islam, and R. M. Nor, "Evaluation of DBSCAN algorithm on different programming languages: An exploratory study," 2021 5th Int. Conf. Electr. Eng. Inf. Commun. Technol. ICEEICT 2021, pp. 1–6, 2021, doi: 10.1109/ICEEICT53905.2021.9667925.

[4] R. Trisminingsih and S. S. Shaztika, "ST-DBSCAN clustering module in SpagoBI for hotspots distribution in Indonesia," Proc. - 2016 3rd Int. Conf. Inf. Technol. Comput. Electr. Eng. ICITACEE 2016, pp. 327–330, 2017, doi: 10.1109/ICITACEE.2016.7892465.

[5] R. Fallahzadeh and H. Ghasemzadeh, "Personalization without user interruption: Boosting activity recognition in new subjects using unlabeled data," Proc. - 2017 ACM/IEEE 8th Int. Conf. Cyber-Physical Syst. ICCPS 2017 (part CPS Week), pp. 293–302, 2017, doi: 10.1145/3055004.3055015.

[6] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander, "OPTICS: Ordering Points to Identify the Clustering Structure," SIGMOD Rec. (ACM Spec. Interes. Gr. Manag. Data), vol. 28, no. 2, pp. 49–60, 1999, doi: 10.1145/304181.304187.

[7] E. Biçici and D. Yuret, "Locally scaled density based clustering," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 4431 LNCS, no. PART 1, pp. 739–748, 2007, doi: 10.1007/978-3-540-71618-1_82.

[8] and J. S. Ricardo J. G. B. Campello, Davoud Moulavi, "Density-based clustering based on hierarchical density estimates," in Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2013, pp. 160–172. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37456-2_14

[9] A. Hinneburg and D. A. Keim, "A General Approach to Clustering in Large Databases with Noise," Knowl. Inf. Syst., vol. 5, no. 4, pp. 387–415, 2003, doi: 10.1007/s10115-003-0086-9.

[10] J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications," Data Min. Knowl. Discov., vol. 2, no. 2, pp. 169–194, 1998, doi: 10.1023/A:1009745219419.

[11] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," Proc. - IEEE Int. Conf. Data Mining, ICDM, pp. 911–916, 2010, doi: 10.1109/ICDM.2010.35.

[12] T. Yuan, Z. Sun, and S. Ma, "Gearbox fault prediction of wind turbines based on a stacking model and change-point detection," Energies, vol. 12, no. 22, 2019, doi: 10.3390/en12224224.

[13] H. P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," Wiley Interdiscip. Rev. Data Min. Knowl. Discov., vol. 1, no. 3, pp. 231–240, 2011, doi: 10.1002/widm.30.

[14] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data," Data Min. Knowl. Discov., vol. 11, no. 1, pp. 5–33, 2005, doi: 10.1007/s10618-005-1396-1.

[15] G. H. Shah, "An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional data sets," 3rd Nirma Univ. Int. Conf. Eng. NUiCONE 2012, pp. 1–6, 2012, doi: 10.1109/NUICONE.2012.6493211.

[16] W. G. Cochran, "Samling Techniques," p. 448, 1977.

[17] E. Hancer and D. Karaboga, "A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number," Swarm Evol. Comput., vol. 32, pp. 49–67, 2017, doi: 10.1016/j.swevo.2016.06.004.

[18] A. Karami and R. Johansson, "Choosing DBSCAN Parameters Automatically using Differential Evolution," Int. J. Comput. Appl., vol. 91, no. 7, pp. 1–11, 2014, doi: 10.5120/15890-5059.

[19] Y. Ren, X. Liu, and W. Liu, "DBCAMM: A novel density based clustering algorithm via using the Mahalanobis metric," Appl. Soft Comput. J., vol. 12, no. 5, pp. 1542–1554, 2012, doi: 10.1016/j.asoc.2011.12.015.

[20] W. Lai, M. Zhou, F. Hu, K. Bian, and Q. Song, "A New DBSCANParameters Determination Method Based on Improved MVO,"IEEE Access, vol. 7, pp. 104085–104095, 2019, doi: 10.1109/ACCESS.2019.2931334.

[21] M. M. R. Khan, M. A. B. Siddique, R. B. Arif, and M. R. Oishe, "ADBSCAN: Adaptive density-based spatial clustering of applications with noise for identifying clusters with varying densities," 4th Int. Conf. Electr. Eng. Inf. Commun. Technol. iCEEiCT 2018, pp. 107–111, 2018, doi: 10.1109/CEEICT.2018.8628138.

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

[23] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial-temporal data," Data Knowl. Eng., vol. 60, no. 1, pp. 208–221, 2007, doi: 10.1016/j.datak.2006.01.013.

[24] C. Tsai and C. Wu, "GF-DBSCAN: A New Efficient and Effective Data Clustering Technique for Large Databases," Proc. 9th WSEAS Int. Conf. Multimed. Syst. signal Process., no. January 2009, pp. 231–236, 2009, [Online]. Available: http://portal.acm.org/citation.cfm?id=1576697.

[25] P. Bholowalia and A. Kumar, "EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN," Int. J. Comput. Appl., vol. 105, no. 9, pp. 975–8887, 2014.

[26] S. J. Yen and Y. S. Lee, "Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset," Lect. Notes Control Inf. Sci., vol. 344, pp. 731–740, 2006, doi: 10.1007/11816492_89.