Analyzing earthquake spread location using a density-based clustering algorithm

Tingting Chen
Computer Science Department
Southern Connecticut State University
Email: chent2@southernct.edu

***Abstract*—This study applies the DBSCAN clustering algorithm to earthquake data, using longitude and latitude to identify spatial patterns of seismic activity. By clustering earthquake occurrences, the method reveals dense regions and isolates noise points, offering insights into the geographical distribution of seismic events. The results highlight clusters associated with subsurface geological features, visualized through a Bouguer gravity anomaly map. The highest silhouette coefficient of 0.5044, achieved with optimized parameters (eps = 0.56, MinPts = 9), demonstrates the clustering method's effectiveness. While the analysis provides valuable insights, limitations include sensitivity to parameter choices and the exclusion of temporal data. Future research will integrate additional features and time-series analysis to improve clustering accuracy and create more precise earthquake distribution maps, aiding in the identification of regions at risk and enhancing public awareness of seismic hazards.**

I. Introduction

Earthquakes [1] occur constantly, with over 20 recorded worldwide in the past 24 hours. In 2024, the number of earthquakes in the US over 70 thousands, causing significant economic losses and potential casualties. Thus , creating earthquake hazard index maps can help the public and government better understand high-risk areas, enabling proactive measures. These maps provide visual data to identify risks and prioritize disaster preparedness, such as constructing earthquake-resistant buildings, developing emergency plans, and promoting public education. They also assist in scientific research and policy-making, optimizing resource allocation and response strategies.

The main focus of this paper is to analyze data using a density-based clustering algorithm [2], evaluate model performance using the silhouette coefficient [3], and ultimately create an earthquake hazard index map. Clustering algorithms are a machine learning technique aimed at grouping data points into clusters based on their similarity or dissimilarity. Within each cluster, the similarity between data points is maximized, while the differences between points in different clusters are maximized. Clustering has broad applications, such as document classification [4], image segmentation [5] and anomaly detection [6].

II. Background and Related Work

Clustering is an essential tool in data mining research and applications. It is the subject of active research in many fields of study, such as computer science, data science, statistics, pattern recognition, artificial intelligence, and machine learning. Several clustering techniques have been proposed and implemented, and most of them successfully find excellent quality or optimal clustering results in the domains mentioned earlier [7].clustering algorithms are primarily categorized into two types:

- Semi-supervised learning uses a training dataset with known cluster labels to help the algorithm learn to assign new data points to clusters, such as in Constrained K-Means or Semi-Supervised Fuzzy C-Means (SSFCM) [8].
- Unsupervised learning algorithms, like K-Means, DBSCAN, and FCM, identify patterns and group similar data points without labeled data, with methods like the elbow method and silhouette score used to determine the optimal number of clusters. [8].

The goal of clustering is subjective. A particular clustering algorithm is oriented toward a particular set of applications. Every algorithm follows a different methodology to specify the closeness of data points. Till date, over 100 clustering algorithms have been proposed and studied. All these can be grouped under five distinct subsets [9]. Fig 1 gives a clear picture of the taxonomy of clustering algorithms.

Classification and clustering are both pattern identification methods in machine learning, but differ in their approach. Classification assigns objects to predefined classes based on their structure, using labeled data to predict the class of unseen observations. It includes binary and multiclass classifi-

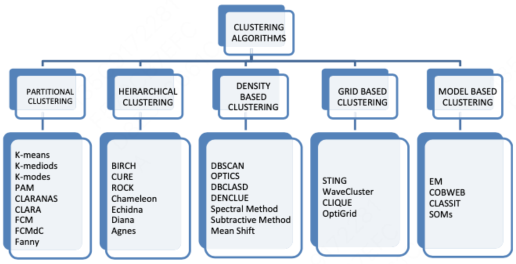

Fig. 1. Categorization of clustering algorithms

cation. An example is a retail company predicting customer segments based on demographic data. In contrast, clustering groups objects based on shared characteristics without predefined labels, identifying similarities and differences between objects. While classification uses class labels, clustering organizes data into clusters based on similarity [10]. As shown in Fig 2.

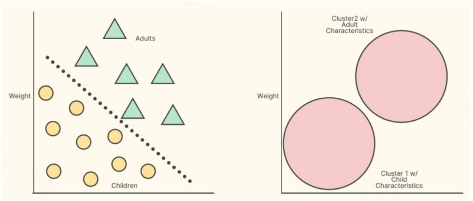
Adults
Weight
Children
Cluster2 w/ Adult Characteristics
Weight
Cluster 1 w/ Child Characteristics

Fig. 2. classification vs clustering

Clustering algorithms group data points based on their characteristics. Partitional clustering (e.g., K-Means, K-Medoids) assigns points to clusters by optimizing distances to centroids, suitable for exclusive cluster membership. Hierarchical clustering (e.g., BIRCH, CURE) creates a nested cluster structure, ideal for hierarchical data. Density-based clustering (e.g., DBSCAN, OPTICS) identifies dense regions, handling noise and outliers, commonly used in spatial data. Grid-based clustering (e.g., STING, CLIQUE) divides data into grids, effective for large datasets. Model-based clustering (e.g., Gaussian Mixture Models) uses statistical models for flexible, probabilistic cluster shapes, useful in complex datasets. These algorithms serve various data types in fields like market segmentation, biology, and geospatial analysis.

K-means and DBScan (Density Based Spatial Clustering of Applications with Noise) are two of the most popular clustering algorithms in unsupervised machine learning. K-Means is a centroid-based algorithm, assuming spherical clusters and requiring the user to specify the number of clusters beforehand, while DBSCAN is a density-based algorithm that can identify clusters of arbitrary shapes and does not require specifying the number of clusters in advance. DBSCAN has a significant advantage when dealing with noise and outliers in the data. In DBSCAN, points that do not belong to any cluster are classified as "noise" and are not assigned to any group. This ability to identify and exclude noise points helps DBSCAN produce more meaningful and robust clusters, especially in real-world datasets where outliers or noise can be prevalent. In contrast, K-means forces all points to belong to some cluster, which can lead to poor results when noise or outliers are present. This makes DBSCAN particularly suitable for datasets where noise and irregularities are common. The comparison between K-Means and DBSCAN is shown in Table I.

TABLE I
Comparison of K-Means and DBSCAN Clustering

Aspect	**K-Means**	**DBSCAN**
Cluster Shape	Assumes spherical clusters	Can detect arbitrary-shaped clusters
Input Parameters	Number of clusters (k)	Epsilon (ϵ) and MinPts
Noise/Outlier Handling	Limited; treats outliers as part of a cluster	Handles outliers explicitly as noise
Performance	Generally faster, but scales poorly with high dimensions	Slower but efficient for low-dimensional data
Cluster Size	Assumes clusters of similar size	Handles clusters of varying sizes
Applications	Well-suited for image segmentation, customer segmentation	Geospatial data, anomaly detection

III. Data Set

The experiment used earthquake data from the BMKG database, covering Indonesia and surrounding regions from January to October 2024. The dataset includes Date, Latitude, Longitude, Depth, and Magnitude. The data processing steps were as follows:

1) I removed website icons and descriptive information that were part of the downloaded dataset.
2) While reading the data, the first row (headers) was excluded, and the columns were renamed to Date, Latitude, Longitude, Depth, and Magnitude.
3) Due to the 10-day data download limit on the website, the original dataset was split into multiple .xlsx files. These were then consolidated into a single .xlsx file.

The final dataset contains 4354 records. Data accuracy was ensured by checking for nulls, outliers, and missing values, with unusable data removed. Table II shows a portion of the cleaned dataset.

TABLE II
Samples of .xlsx earthquake data

Date	Latitude	Longitude	Depth	Magnitude
2024-08-19	-9.77	118.47	44	3.44
2024-08-19	2.42	98.99	136	4.15
2024-08-19	-8.85	110.10	10	3.51
2024-08-19	-10	123.74	28.8	4.36
2024-08-20	0.45	123.65	244.9	4.02
2024-08-21	0.32	124	210.2	8.92

IV. Methods

A. Density-based Algorithm

Density-based algorithms [11] are clustering techniques that identify clusters as dense regions in the data space, separating them from areas of low density. These methods are particularly effective for detecting clusters of arbitrary shapes and handling noise or outliers. The core idea revolves around defining clusters based on the density of data points within a specified neighborhood radius (Epsilon, ϵ) and a minimum number of points (MinPts) within that neighborhood. Such algorithms are widely used in geospatial analysis, anomaly detection, and bioinformatics due to their adaptability to complex datasets.

There are three types of points in a density-based clustering approach: core points, border points, and noise points. Core points are those with at least the specified minimum number of neighbors (MinPts) within a defined radius (ϵ), forming the dense "core" of a cluster. Border points, which lie within the ϵ-neighborhood of a core point, have fewer neighbors than MinPts and often serve as the boundary of a cluster. Notably, a border point may belong to more than one cluster if it falls within the neighborhoods of multiple core points. Noise points, on the other hand, do not belong to any cluster as they fall outside the ϵ-neighborhoods of all core points. Given a dataset of D objects, all core points can be identified using ϵ and MinPts, with clusters emerging from the dense regions formed by connecting core points and their associated neighborhoods. Noise points help distinguish meaningful clusters by separating out sparse or irrelevant data. Figure 3 shows the defination between points:

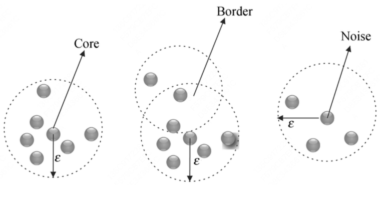

Fig. 3. DBSCAN: core, border, and noise points.

There are several terms in DBSCAN:

1) **Directly density-reachable**: For core object q and object p, it says that p is directly density-reachable from q (with ϵ and MinPts) if p is in the ϵ-neighborhood of q.
2) **Density-reachable**: p is density-reachable from q (with ϵ and MinPts in D) if there is a chain of objects $p_1, \ldots, p_n$ such that $p_1 = q$, $p_n = p$, and p_{i+1} are directly density-reachable from p_i with ϵ and MinPts, for $1 \leq i \leq n$, $p_i \in D$.
3) **Density-connected**: Two objects $p_1, p_2 \in D$ are density-connected (with ϵ and MinPts) if there are objects $q \in D$ such that p_1 and p_2 are density-reachable from q with ϵ and MinPts.

Figure 4 is an example of density-reachability and density-connectivity for a certain ϵ, represented by the circle radius, assuming MinPts = 3. Points m, p, o, and r are core objects because each one is in an ϵ-neighborhood that contains at least three points. Object q is directly density-reachable from m. Object m is directly density-reachable from p and vice versa. Object q is (indirectly) density-reachable from p because q is directly density-reachable from m and m is directly density-reachable from p. However, p is not density-reachable from q because q is not a core object. Similarly, r and s are density-reachable from o, and o is density-reachable from r. Therefore, o, r, and s are all density-connected [2].

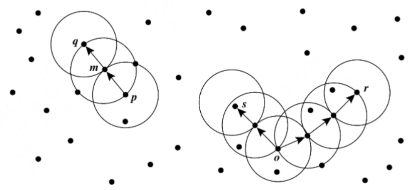

Fig. 4. Density-reachability and density-connectivity

DBSCAN finds clusters by tracing the clusters, which involves examining the ϵ-neighborhood (Eps-neighborhood) of each point in the database. If the ϵ-neighborhood of point p contains more than MinPts, a new cluster with p as the core object is created. Then, DBSCAN iteratively collects density-reachable objects directly from the core object, which may involve merging several density-reachable clusters. The sequence of the DBSCAN algorithm is as follows:

1) Choose the initial point p randomly,
2) Determine ϵ and MinPts to take all points that are density-reachable to point p,

3) If p is the core point, then a cluster is formed,
4) If p is a border point, there is no density-reachable relation of p, and DBSCAN will visit the next point in the database,
5) Continue processing until all points have been processed,
6) The result obtained does not depend on the order of the processed points taken.

Workflow as Figure 5.

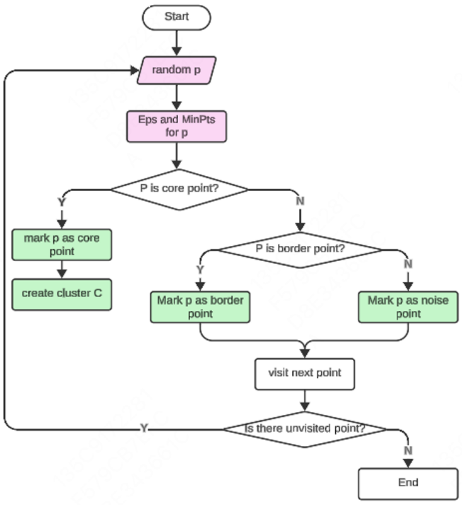

Fig. 5. DBSCAN workflow

B. *Evaluation*

Silhouette coefficient is a metric used to evaluate the quality of clustering in computer science. It measures the coherence of clusters, with a higher coefficient indicating more coherent clusters. The coefficient ranges from -1 to 1, with values close to +1 indicating that a sample is far away from neighboring clusters, and negative values suggesting that samples may have been assigned to the wrong cluster. The coefficient is calculated based on cluster cohesion and cluster separation, which represent the average distances between instances and data points within and between clusters, respectively. The Silhouette Coefficient $s(i)$ for a point i is given by the formula:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

where:

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} d(i, j)$$

is the average distance from point i to other points in the same cluster C_i, and

$$b(i) = \min_{k \neq C_i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

is the average distance from point i to all points in the nearest cluster C_k.

V. Experimental Setup

A. *Features Selection*

In the earthquake analysis, longitude and latitude coordinates are selected as the main clustering features for DBSCAN analysis. These geographic features help group earthquake events based on their spatial proximity, identifying potential clusters of seismic activity. By using these coordinates, the analysis focuses on the geographical distribution of earthquakes, facilitating the identification of patterns and clusters in the data.

B. *Parameters Selection*

In the earthquake analysis, parameter selection is crucial for effective clustering using DBSCAN. The two primary parameters for DBSCAN are eps (the maximum distance between two points to be considered as neighbors) and MinPts (the minimum number of points required to form a dense region or cluster). These parameter choices are critical for achieving accurate clustering results, as a small eps value may result in too many small clusters or noise points, while a large value may lead to merging distinct clusters. Similarly, a low MinPts can lead to smaller, less reliable clusters, while a higher value ensures more robust clusters by requiring more points. The selected parameters help balance the trade-off between detecting real clusters and minimizing noise points.

In this experiment, the maximum, minimum, and median Euclidean distances between the points are first calculated to determine the approximate range for the eps parameter, as values that are either too high or too low are unsuitable. In DBSCAN, the value of MinPts is determined using the K-distance graph, where the value of k is set to (2 * dimension - 1). Since both longitude and latitude are used, k = 3. The appropriate eps values are identified where the graph exhibits elbow points. The resulting K-distance graph is shown in Fig. 6. From the graph, we observe that the main inflection point occurs around eps = 0.1. For thoroughness, we calculated the silhouette coefficients(III) for all values of eps with a step size of 0.05, ranging from 0.01 to 0.6, and for MinPts with a step size of 1, ranging from 3 to 12. The results show that when eps = 0.1, the silhouette coefficient is not optimal. The highest silhouette coefficient of 0.5044 occurs when eps = 0.56 and MinPts = 9. Therefore, we use eps =

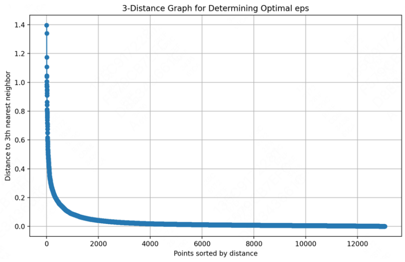

Fig. 6. K-distance diagram

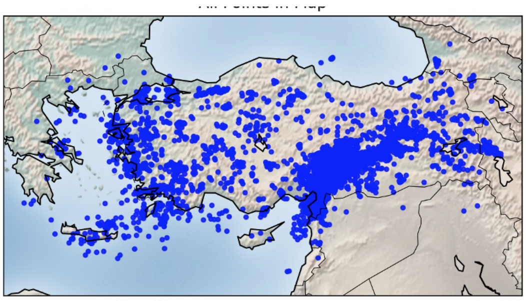

Fig. 7. All points in map

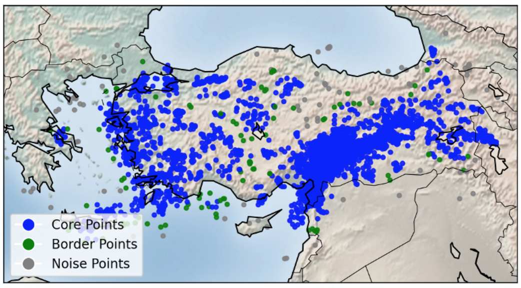

Fig. 8. Core, Border, Noise Points in map

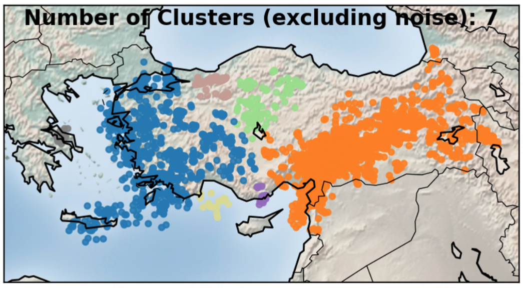

Fig. 9. Clusters

0.56 and MinPts = 9 as the input values for the DBSCAN algorithm.

TABLE III
Silhouette coefficient values under different eps and MinPts

eps	MinPts	silhouette score	real cluster num
0.11	3	-0.0599	144
		...	
0.11	11	0.0866	34
		...	
0.56	3	-0.1101	10
		...	
0.56	9	0.5044	7
		...	
0.56	11	0.4736	8

VI. Results

The fig7 shows the location of all points (earthquake data) on a map, giving a visual overview of the data distribution. The fig8 categorizes the points into core, border, and noise points based on the DBSCAN clustering results, illustrating how different types of points are spatially distributed and their relationship to the clustering algorithm.

The fig9 visually represents the clusters identified by the DBSCAN algorithm, with each cluster assigned a distinct color. DBSCAN, being a density-based algorithm, groups points that are closely packed together and separates them from points that are sparse, which are considered as noise (not part of any cluster). The plot excludes noise points (labeled as -1) and visualizes only the clusters. This plot helps to understand the spatial distribution of different clusters within the data. The colored clusters highlight regions with a high density of earthquake occurrences, suggesting areas where seismic activity may be more concentrated. The separation of noise points is important because it indicates areas that do not conform to any discernible seismic patterns, which could be outliers or regions with very low activity. The identification of these clusters provides a clearer picture of the distribution of earthquake activity across the geographical area, which could be useful for targeted analysis of seismic hotspots or for further investigation into the factors influencing seismic events in those regions.

The fig10 visualizes the Bouguer gravity anomaly [12] across the region using an interpolated grid. The anomaly values are calculated based on the depth and magnitude of each earthquake, and are visualized using a color gradient from green (low anomaly) to red (high anomaly). This plot uses the griddata function to interpolate the anomaly values over a regular grid, which smooths the data and provides a continuous surface view of the anomalies. The Bouguer gravity anomaly map offers insights into the geological characteristics of the region, with higher anomaly values potentially indicating areas of significant underground structures, such as fault lines or tectonic plate boundaries, which could influence earthquake occurrence. The map's color gradient helps in visually identifying regions of higher gravity anomalies, which may correlate with areas of increased seismic activity. By analyzing the spatial relationship between the

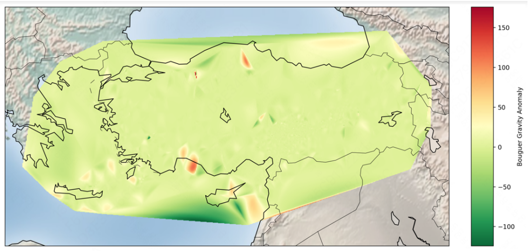

Fig. 10. Bouguer gravity anomaly map

clusters (Plot 3) and the gravity anomalies (Plot 4), one can hypothesize potential links between the seismic activity and subsurface geological features. This analysis can guide future research into understanding how geological formations may contribute to earthquake generation in specific regions, offering valuable information for seismic risk assessments.

VII. Conclusion and Future Work

This study demonstrates the effectiveness of the DBSCAN algorithm in clustering earthquake data based on spatial density. Currently, the analysis is subject to potential influences from parameter selection and limitations due to the lack of temporal and geological background data. Future research could incorporate additional features to improve clustering accuracy and provide a more comprehensive understanding of seismic dynamics.

- Bouguer Gravity Anomaly Model Optimization [13]: The current calculation of Bouguer gravity anomaly uses a simple multiplication model, but other influencing factors could be introduced.
- Time Series Analysis [14]: Since earthquakes occur over time, analyzing the temporal patterns of events is very important.
- Anomaly Detection [15]: Applying anomaly detection algorithms could help identify potential unusual earthquake events.

Code is available at https://github.com/tingting0523/CSC521-Algorithms.git.

References

[1] P. L. I. M. G. L. Roberto Basile, Luisa Giallonardo and R. Persio, “The local labour market effects of earthquakes,” *Regional Studies*, vol. 58, no. 1, pp. 91–104, 2024. [Online]. Available: https://doi.org/10.1080/00343404.2023.2187045

[2] M. Bariklana and A. Fauzan, “Implementation of the dbscan method for cluster mapping of earthquake spread location,” *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259485912

[3] V. Siless, S. Medina, G. Varoquaux, and B. Thirion, “A comparison of metrics and algorithms for fiber clustering,” in *2013 International Workshop on Pattern Recognition in Neuroimaging*, 2013, pp. 190–193.

[4] I. Pauletic, L. N. Prskalo, and M. B. Bakaric, “An overview of clustering models with an application to document clustering,” in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019, pp. 1659–1664.

[5] H. Mittal, A. C. Pandey, M. Saraswat, S. Kumar, R. Pal, and G. Modwel, “A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets,” *Multimedia Tools Appl.*, vol. 81, no. 24, p. 35001–35026, Oct. 2022. [Online]. Available: https://doi.org/10.1007/s11042-021-10594-9

[6] I. Syarif, A. Prugel-Bennett, and G. Wills, “Unsupervised clustering approach for network anomaly detection,” in *Networked Digital Technologies*, R. Benlamri, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 135–145.

[7] A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, and A. A. Akinyelu, “A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects,” *Engineering Applications of Artificial Intelligence*, vol. 110, p. 104743, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095219762200046X

[8] H. Yin, A. Aryani, S. Petrie, A. Nambissan, A. Astudillo, and S. Cao, “A rapid review of clustering algorithms,” *ArXiv*, vol. abs/2401.07389, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:266999735

[9] A. Ghosal, A. Nandy, A. K. Das, S. Goswami, and M. Panday, “A short review on different clustering techniques and their applications,” in *Emerging Technology in Modelling and Graphics*, J. K. Mandal and D. Bhattacharya, Eds. Singapore: Springer Singapore, 2020, pp. 69–83.

[10] W. Z. T. Tareq and M. Davud, “Chapter 20a - classification and clustering,” in *Decision-Making Models*, ser. Uncertainty, Computational Techniques, and Decision Intelligence, T. Allahviranloo, W. Pedrycz, and A. Seyyedabbasi, Eds. Academic Press, 2024, pp. 351–359. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780443161476000244

[11] W.-K. Loh and Y.-H. Park, “A survey on density-based clustering algorithms,” in *Ubiquitous Information Technologies and Applications*, Y.-S. Jeong, Y.-H. Park, C.-H. R. Hsu, and J. J. J. H. Park, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 775–780.

[12] D. A. Chapin, “The theory of the bouguer gravity anomaly; a tutorial,” *The Leading Edge*, vol. 15, no. 5, pp. 361–363, 05 1996.

[13] Y.-T. Lo, K.-E. Ching, H.-Y. Yen, and S.-C. Chen, “Bouguer gravity anomalies and the three-dimensional density structure of a thick mudstone area: A case study of southwestern taiwan,” *Tectonophysics*, vol. 848, p. 229730, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0040195123000288

[14] A. Amei, W. Fu, and C.-H. Ho, “Time series analysis for predicting the occurrences of large scale earthquakes,” 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:52263718

[15] M. Çelik, F. Dadaşer-Çelik, and A. Dokuz, “Anomaly detection in temperature data using dbscan algorithm,” in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 2011, pp. 91–95.