

Article

Application of Three-Dimensional Hierarchical Density-Based Spatial Clustering of Applications with Noise in Ship Automatic Identification System Trajectory-Cluster Analysis

Shih-Ming Wang ¹, Wen-Rong Yang ^{2,*}, Qian-Yi Zhuang ², Wei-Hong Lin ², Mau-Yi Tian ², Te-Jen Su ^{2,*} and Jui-Chuan Cheng ²

¹ Department of Computer Science and Information Engineering, Cheng Shiu University, Kaohsiung City 833301, Taiwan; k1115@gcloud.csu.edu.tw

² Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung City 807618, Taiwan; j111252102@nkust.edu.tw (Q.-Y.Z.); i110152102@nkust.edu.tw (W.-H.L.); j110252107@nkust.edu.tw (M.-Y.T.); eagle@nkust.edu.tw (J.-C.C.)

* Correspondence: wryang@nkust.edu.tw (W.-R.Y.); sutj@nkust.edu.tw (T.-J.S.)

Abstract: Clustering algorithms are widely used in statistical data analysis as a form of unsupervised machine learning, playing a crucial role in big data mining research for Maritime Intelligent Transportation Systems. While numerous studies have explored methods for optimizing ship trajectory clustering, such as narrowing dynamic time windows to prevent errors in time warp calculations or employing the Mahalanobis distance, these methods enhance DBSCAN (Density-Based Spatial Clustering of Applications with Noise) by leveraging trajectory similarity features for clustering. In recent years, machine learning research has rapidly accumulated, and multiple studies have shown that HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) outperforms DBSCAN in achieving accurate and efficient clustering results due to its hierarchical density-based clustering processing technique, particularly in big data mining. This study focuses on the area near Taichung Port in central Taiwan, a crucial maritime shipping route where ship trajectories naturally exhibit a complex and intertwined distribution. Using ship coordinates and heading, the experiment normalized and transformed them into three-dimensional spatial features, employing the HDBSCAN algorithm to obtain optimal clustering results. These results provided a more nuanced analysis compared to human visual observation. This study also utilized O notation and execution time to represent the performance of various methods, with the literature review indicating that HDBSCAN has the same time complexity as DBSCAN but outperforms K-means and other methods. This research involved approximately 293,000 real historical data points and further employed the Silhouette Coefficient and Davies–Bouldin Index to objectively analyze the clustering results. The experiment generated eight clusters with a noise ratio of 12.7%, and the evaluation results consistently demonstrate that HDBSCAN outperforms other methods for big data analysis of ship trajectory clustering.

Keywords: HDBSCAN; AIS; ship trajectories; clustering



Academic Editor: José A. Orosa

Received: 5 January 2025

Revised: 22 February 2025

Accepted: 26 February 2025

Published: 28 February 2025

Citation: Wang, S.-M.; Yang, W.-R.; Zhuang, Q.-Y.; Lin, W.-H.; Tian, M.-Y.; Su, T.-J.; Cheng, J.-C. Application of Three-Dimensional Hierarchical Density-Based Spatial Clustering of Applications with Noise in Ship Automatic Identification System Trajectory-Cluster Analysis. *Appl. Sci.* **2025**, *15*, 2621. <https://doi.org/10.3390/app15052621>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The ship's trajectory refers to the positional information transmitted by a specific system during navigation. This system is called the Automatic Identification System (AIS). The primary function of the AIS, designed for collision avoidance, is an integrated hardware device that uses Very-High-Frequency (VHF) radio waves to broadcast and

receive information. It features navigation and assistance functions, providing dynamic information about the vessel. The AIS serves as a crucial communication system for land-based maritime management and acts as a key source of data for the future development of autonomous maritime navigation. Furthermore, the AIS offers advanced functionalities, including maritime traffic flow analysis, warnings for navigational risks such as course deviations and grounding, and applications in maritime incident investigations. It plays a vital role in the ongoing development of the Maritime Intelligent Transportation System (M-ITS). To further maximize the value of AIS data, various algorithms are applied for data mining, enabling rapid and accurate processing of vast amounts of information. More importantly, these algorithms can be integrated with machine learning in the future, allowing for the development of AI models to optimize the processing and analysis of maritime traffic information.

The ship's trajectory represents the vessel's navigation path and behavior. Clustering analysis can effectively classify these trajectory groups using algorithms, which helps assess potential risks to navigation safety. The trajectory clustering algorithm proposed in this study, through feature analysis and pattern extraction, yields clustering results that include both grouped and scattered classifications, which can indeed contribute to navigation safety. For example, the scattered data help identify abnormal navigation trajectories, while the clustering results facilitate traffic density calculations. Furthermore, when combined with other maritime information, they can also aid in making optimized routing decisions. Analysis of trajectories produces grouping results that include multiple clusters of trajectories along with scattered noise. The clustered trajectories help analyze traffic density, while the scattered trajectories can explain route deviations or serve as risk warnings. The research results of trajectory clustering offer critical insights for ensuring navigational safety and optimizing maritime decision-making. Consequently, shipping companies and maritime traffic authorities may find this technology of great interest. These suggestions hold significant importance for guiding the future development of the technology. This contributes to the future development of intelligent maritime management technologies. Clustering algorithms are unsupervised methods that can aggregate data into different categories, known as clusters, without prior knowledge. These algorithms are particularly effective for datasets where the number of clusters cannot be predicted in advance. Many clustering algorithms have been proposed in the literature, including K-means, DBSCAN [1], and HDBSCAN [2]. DBSCAN is designed based on the assumption of homogeneous data distribution. In such cases, DBSCAN and HDBSCAN perform similarly. However, for heterogeneous distributions, DBSCAN relies on two parameters, epsilon (*eps*) and minimum sample size (*minPts*), to calculate the clustering thresholds, which require iterative tuning to achieve optimal results. HDBSCAN by contrast, automatically determines *eps* through density gradients, eliminating the need for manual adjustment. This approach improves clustering performance and noise identification through density stratification. After a thorough comparison, we chose to focus on the application of HDBSCAN, primarily due to its outstanding capability in handling large amounts of noise and big data. This algorithm has been widely applied in fields such as medical image analysis, remote sensing data processing, social network analysis, public health, security, and transportation, with numerous studies demonstrating its exceptional performance. This research is the first to apply HDBSCAN to the clustering analysis of maritime vessel trajectory traffic, and we hope to contribute to advancements in this field.

This study focuses on historical data from the waters near Taichung (TC) Port in Taiwan. To achieve optimal trajectory clustering analysis, it is essential to understand AIS data and the characteristics of vessel navigation deeply. The data must be cleaned, and features must be selected according to these characteristics. Section 2 of this paper

reviews the related literature on trajectory data preprocessing and clustering applications. Section 3 describes the characteristics of AIS data and the preprocessing methods in detail, including data cleaning, trajectory segmentation, and simplification. The latter part introduces clustering algorithms and explains the selection and optimization of clustering features and feature normalization. Section 5 presents the results of the experimental data, detailing data processing, analyzing the clustering results of ship trajectories and the causes of noise formation, and finally comparing the clustering performance of HDBSCAN and DBSCAN using evaluation metrics. Finally, Section 6 summarizes the contributions and limitations of this study and identifies directions for future research.

2. Literature Review

Many papers have already explored related data cleaning and clustering analysis methods. For instance, G. Yuan et al. (2011) proposed the TOD-SS (Trajectory Outlier Detection algorithm based on Structural Similarity) algorithm, which detects outliers using an angle threshold based on the structural similarity of trajectory segments [3]. K. Lin et al. (2016) evaluated trajectory outliers using speed and time thresholds [4]. Early et al. (2020) focused on calculating the drift of ocean buoys, using interpolation to smooth the actual GPS trajectories [5]. They also used multiple moving interpolation smoothing methods and distance thresholds for fast calculation. The methods for noise elimination include the Kalman filter, a highly efficient recursive filter. It calculates the current state estimate based on the previous state estimate and the current observation and can be used to assess noise outliers and predict future trajectories. The Gaussian Mixture Model (GMM) [6], as discussed by R. Laxhammar et al. (2014), is commonly used to filter out background noise. Several trajectory compression or simplification methods exist, which reduce sampling points based on the maximum distance threshold epsilon between points and line segments. A classic method is the Douglas–Peucker algorithm (Douglas et al., 1973) [7]. In recent years, with significant advancements in computing power, many studies have discussed and improved the DP method [8,9]. S. K. Zhang et al. (2016) proposed a measurement of the DP threshold, verified through plotting [10]. L. Zhao et al. (2019) determined the appropriate DP threshold based on changes in trajectory shape and used it to compress trajectories before calculating the Dynamic Time Warping (DTW) distance matrix [11]. W. Cao et al. (2017) proposed the DOTS (Directed acyclic graph-based Online Trajectory Simplification) method [12], which uses Synchronized Euclidean Distance (SED) as a substitute distance calculation to improve computational convenience, enabling real-time operations. In the same year, Q. Meng et al. (2017) proposed the OPW-TR (open-window time ratio) method to reduce errors in SED distance calculations [13]. Other literature compares the speed of DP and various distance calculations [14] (Jonathan Muckell et al. (2010)). The Mahalanobis distance (DM) calculation method, involving vector calculations of the covariance matrix, was proposed as early as 1936 by Indian statistician Mahalanobis. Additionally, during navigation, the Dead Reckoning (DR) method, which utilizes previously determined positions to calculate the current location of a moving object, is often employed. The Monte Carlo Localization (MCL) method, a statistical analogy method, estimates and resamples particle distributions in space based on Recursive Bayesian estimation theory. The most common risk function is Mean Squared Error (MSE). J. Muckell (2022) introduced the SQUISH method [15], a real-time algorithm for trajectory compression. It calculates the SED distance of trajectory segments. It removes the maximum value, adds new points to fill the gaps, maintains a fixed data count, and repeats the process to achieve the goal of data compression. Clustering algorithms are among the most valuable methods in the field of computational intelligence, frequently seen in AI applications. Ester et al. (1996) proposed DBSCAN, a density-based algorithm for spatial clustering. In DBSCAN, two critical pa-

parameters are epsilon, the distance threshold, and MiniPts, the minimum number of points required to form a cluster. Many subsequent scholars have proposed comparative or improved versions, such as the comparison between KNN and DBSCAN [16] (Yu et al. (2005)); dynamically computed DMDBSCAN [17] (Mohammad et al. (2013)); DSets-DBSCAN [18] (Hou et al. (2016); image segmentation using pixels as sets); Soft DBSCAN [19] (Smiti et al. (2013); combining DBSCAN with fuzzy set theory); and GMDDBSCAN [20] (Chen et al. (2006)) and HDBSCAN [21] (Lianhui Wang et al. (2021)), based on spatial indexing and grid technology.

In recent years, numerous spatial clustering algorithms based on DBSCAN have emerged. For example, OPTICS, Deng et al. (2015) [22], enhanced density processing capabilities, and PDBSCAN [23], Jiang et al. (2011), handled data of varying densities through partitioning, laying the groundwork for the development of HDBSCAN. Additionally, HPDBSCAN, Götz et al. (2015) [24], introduced parallel computing to accelerate big data processing; Fuzzy-DBSCAN, Ienco et al. (2018) [25], incorporated fuzzy theory to allow for fuzzy cores and overlapping boundaries; ST-DBSCAN, Trisminingsih et al. (2016) [26], added a temporal dimension; DBSCAN+, Xie et al. (2021) [27], strengthened noise and boundary handling; and K-DBSCAN, Gholizadeh et al. (2021) [28], utilized K-means for preliminary classification before applying DBSCAN for finer segmentation to speed up computation. Among these numerous algorithms, the literature on HDBSCAN has accumulated the highest citation count, indicating its significant influence in both academia and practical applications, which is an important consideration for us in choosing HDBSCAN.

3. Data and Methods

The AIS data structure is quite complex, consisting of 27 types of messages. Ship trajectories are recorded in messages of types 1, 2, 18, and 19, known as Position Report messages. Shore-based receiving stations capture these time-series data. Position Report messages contain information about the ship and its navigation, such as the seven-digit code issued by the International Maritime Organization (IMO) representing the ship's identifier and the 9-digit Maritime Mobile Service Identity (MMSI) issued by regional authorities. They also include the ship's position coordinates (longitude and latitude), speed over ground (SOG), course over ground (COG), heading (HDG), ship type code, and the time the base station received the message. In Taiwan, due to regulatory requirements, fishing vessels are required to install Vessel Monitoring Systems (VMSs), Digital Selective Calling (DSC) radios, and Emergency Position Indicating Radio Beacons (EPIRB). As a result, the data for fishing vessels often need to be completed or even contain null values. Furthermore, fishing vessels are typically distributed near shore and have disordered trajectories, so filtering out fishing vessels from the analysis is recommended. These Position Report messages represent the historical trajectories of ships at sea, providing valuable data for research and analysis. The data proportion for other vessels, such as passenger ferries, speedboats, and workboats, is less than 5%. Therefore, this study only uses the trajectories of cargo ships and tankers as research data. This section details the data used and the research area for this study. The study focuses on the busy waters west of Taichung Port (TC) in central Taiwan, with a spatial distribution ranging from 119.5 to 121 degrees east longitude and 23.5 to 25 degrees north latitude. This area connects the Taiwan Strait's north and south shipping lanes, linking Northeast Asia with Southeast Asia. The area has busy north-south shipping routes, as well as east-west routes to and from TC and Kinmen (KM) and the tracks along the coast of mainland China, as shown in Figure 1.

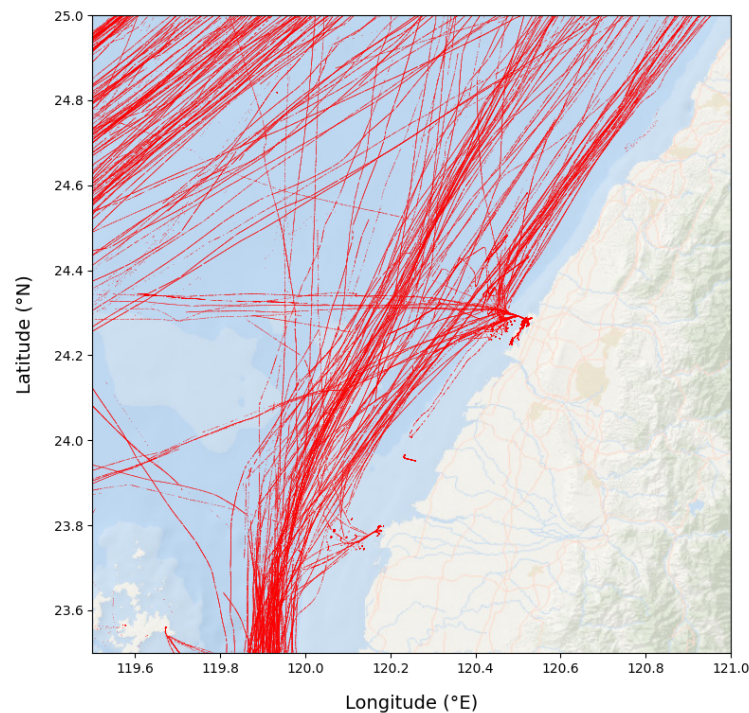


Figure 1. This map shows the coast of central Taiwan. There are many red lines on the map, representing the tracks of ships.

Due to the vast amount of AIS ship trajectory data—take July 2020 as an example, when the total number of entries around Taiwan reached 320 million—employing fast and effective data cleaning methods is crucial for subsequent clustering analysis. Therefore, the experimental process in this work is divided into two stages: The first stage involves data cleaning, including trajectory segmentation, trajectory compression, and speed filtering. The second stage focuses on clustering analysis, involving the selection and normalization of three-dimensional features and an introduction to trajectory clustering algorithms. The overall research process framework is illustrated in Figure 2, and the details are explained in the following two paragraphs.

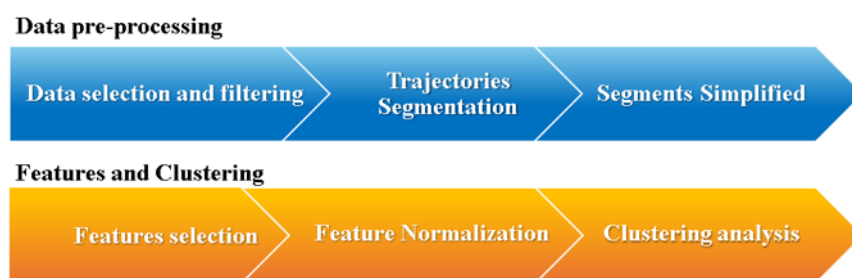


Figure 2. Research Process Framework.

3.1. Data Pre-Processing

Data preprocessing can be divided into data cleaning, data integration, and data transformation. The main goal is to optimize the data quality for clustering analysis by handling the data based on its reception characteristics. For instance, the AIS, a VHF broadcast communication system, may encounter issues with duplicate data from multiple shore-based receiving stations. The NMEA 0183 transmission baud rate is typically 4800, which needs to be higher to meet the demands of large data transmission. Even the

more advanced and faster NMEA2000 can experience congestion and delays due to high transmission volumes. Some regionally active, non-international vessels do not apply for an IMO number and instead use an MMSI code as the unique identifier. However, multiple vessels may have the same MMSI in overlapping regions, necessitating debugging or trajectory partitioning. Additionally, due to the collision avoidance design of the AIS, the update frequency varies with the ship's speed according to ITU-R M.1371-1 [29]. It is a recommendation standard developed by the International Telecommunication Union (ITU) for the Automatic Identification System (AIS), in which details the reporting intervals of AIS devices under different ship dynamic conditions, as shown in Table 1. The faster the ship's speed, the higher the broadcast update frequency, ranging from 2 s to 3 min. When a vessel is anchored or moored, if its speed exceeds 3 knots, the update frequency is 10 s; if the speed is below 3 knots, the update frequency drops to 3 min. The largest gap in time intervals occurs around the speed threshold of 3 knots. Therefore, a speed of 3 knots serves as a critical reference point for distinguishing the movement speed of a vessel in an anchored or moored state.

Table 1. ITU-R M.1371-1.

Ship's Dynamic Conditions	Reporting Interval
Ship at anchor and not moving faster than 3 knots	3 min
Ship at anchor and moving faster than 3 knots	10 s
Ship 0–14 knots	10 s
Ship 0–14 knots	3 1/3 s
Ship 14–23 knots	6 s
Ship 14–23 knots	2 s
Ship high than 23 knots	2 s

In Section 2, many studies related to trajectory simplification methods have been listed, and it is explained that simplified trajectories do not affect clustering analysis, such as in the work of Yuan et al. (2011) [3]. Therefore, the continuity of the AIS trajectory is utilized to complete simple trajectory simplification by setting the COG angle change threshold, thereby improving calculation efficiency. The AIS historical data used in this work were filtered to exclude fishing vessels and unknown ship types, leaving primarily cargo and oil tankers. The total amount of data in the region was approximately 170 million, with around 5.4 million daily records. There were 3101 unique MMSI codes, representing 3101 ships around Taiwan daily. After the selected study area was narrowed down, the remaining data amounted to 557,000 records, with 432 unique MMSI codes.

3.2. Features and Clustering

In analyzing trajectory clustering data, both COG and longitude and latitude are crucial features. Therefore, this study will use three-dimensional features for clustering analysis. In theoretical statistics, parametric normalization can often lead to pivotal quantities—functions whose sampling distribution does not depend on the parameters—and to ancillary statistics—pivotal quantities that can be computed from observations without knowing the parameters. Normalization ensures that all features contribute equally during computation by adjusting and aligning probability distributions. After normalizing the three-dimensional features, they are transformed into X, Y, and Z, with values ranging between 0 and 1.

Before applying the algorithm, analyzing Figure 1, using the human visual clustering method, the trajectories group can be roughly divided as follows (see Table 2): (1) The shape of ship trajectory clustering in the TC area is like a funnel. North of 23.6° N, the tracks are more dispersed, but the southern ones are more concentrated. The shape of the track

clusters is related to the depth and topography of the water. (2) The path between the TC and KM A cluster of east–west trajectories. (3) Tug boats frequently operate at offshore wind farms. (4) Traditional fishing activities at Changhua Fishing Port (CH) lead to complex sea routes. (5) A cluster of trajectories along the coast of mainland China also shows NE–SW distribution.

Table 2. Expected results comparison for the 2 methods.

Human Visual	HDBSCAN
Ship at anchor and moving faster than 3 knots	cluster 1 heading NE cluster 2 heading NWN and turning to KN cluster 4 heading SW terminating at TC cluster 7 heading SW from NE or from TC
East–west between TC–KM	cluster 5 heading W from TC cluster 6 coming out from TC
Near mainland China	cluster 0 heading NE cluster 3 heading SW

The goal is to confirm these observations further using algorithms. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that is particularly effective for arbitrary-shaped clusters and for handling noisy data. DBSCAN does not require a pre-specified number of clusters and can identify noise points, clustering based on density. The algorithm identifies clusters in a given dataset using two key parameters: the maximum distance epsilon to form a neighborhood and the minimum number of points required to form a cluster (*minPts*). However, since DBSCAN assumes that all particles have a similar density, it may not produce good clustering results when the data distribution does not clearly distinguish between dense and sparse areas.

HDBSCAN Algorithm Data Processing Steps: According to Campello, Moulavi, and Sander (2015) [30], the HDBSCAN algorithm is introduced in Section 4.1, and the data processing steps of the HDBSCAN algorithm are explained in detail.

Step 1: Core Distance Calculation

The core distance is calculated by determining the distance between all data points, usually using Euclidean distance or other distance metrics. The core distance for each data point p is defined as the distance from p to its k nearest neighbor, as represented by Equation (1):

$$\text{CoreDistance}(p) = d(p, q_k) \quad (1)$$

where q_k is the k nearest neighbor of p .

Step 2: Constructing the Minimum Spanning Tree (MST) Based on the Distance Matrix

The MST is a subgraph of a connected undirected graph that includes all the vertices of the original graph. It has a tree-like structure, clearly representing branches and hierarchy. The thickness of the branches represents the magnitude of the values. The visualization of the tree structure is a key feature of the HDBSCAN algorithm, helping users understand the structure and persistence of the clusters.

Step 3: Calculating the Stability of Clusters

Since hierarchical clustering is used, the more frequently a cluster appears at different levels, the higher its stability. The stability $S(C)$ of a cluster is calculated using the sum of the differences between the maximum core distance $d_i^{\max}$ and the minimum core distance $d_i^{\min}$ of each cluster at different levels, as shown in the following Equation (2):

$$S(C) = \sum_i (d_i^{\max} - d_i^{\min}) \quad (2)$$

where S is the stability score, C represents the cluster, and d is the core distance. Clusters that represent noise points are marked as -1 , indicating that these points do not belong to any cluster. Finally, each data point x is labeled as belonging to a cluster C or as noise, represented as Equation (3):

$$\text{label}(x_i) = \begin{cases} C_j & \text{if } x_j \text{ belongs to cluster } C_j \\ \text{noise} & \text{otherwise} \end{cases} \quad (3)$$

Based on the steps above, the foundation of the HDBSCAN algorithm is constructed, enabling it to hierarchically and effectively handle clustering analysis related to big data.

4. Computational Efficiency

Regarding the description of computational efficiency, according to Sedgewick and Wayne (2011) [31], (Section 3.3, the first paragraph of that section), where it is explained as a standard way to quantify the time and space complexity of algorithms, O notation is used to describe the time and space complexity of algorithms, providing a standard method for evaluating algorithm performance. According to Campello et al. (2015), the time complexity of HDBSCAN can be expressed as $O(n \cdot \log n)$. While the worst-case complexity is $O(n^2)$, it can achieve $O(n \cdot \log n)$ when spatial indexing is used. The time complexity of K-means is $O(n \cdot k \cdot i)$, where k is the number of clusters and i is the number of iterations. When the dataset is very large and the number of clusters increases, the computational cost rises rapidly. In addition, the calculation of similar trajectories, such as the Mahalanobis distance, involves vector multiplication and summing squares, which requires covariance matrix computation. For n samples and p features, the time complexity is $O(n^2 \cdot p^2)$, which clearly requires more. The Douglas–Peucker algorithm assumes the curve has n points. In the worst case, its time complexity is $O(n \cdot \log n)$, and in the best case, it is $O(n)$. Therefore, it is generally expressed as $O(n \cdot \log n)$. The OPTICS algorithm's main time cost comes from calculating core distances and sorting. Assuming quicksort is used, the time complexity is $O(n \cdot \log n)$. The SQUISH algorithm has a time complexity of $O(n)$ for interpolation, but if shape fitting or data compression is involved, it can reach $O(n^2)$ or higher.

We understand that execution time is a key indicator in big data applications. However, we also recognize that execution time is significantly influenced by hardware specifications, data size, and parameter settings. In this experiment, the data volume is 293,000 records. By performing multiple clustering algorithm tests, we obtained the average execution times. Specifically, the average time for HDBSCAN over 10 runs was 38.37 s, DBSCAN was 213.16 s, K-means was 336.74 s, and OPTICS took the longest, with only 2 runs yielding an average time of 3091.87 s, as shown in Table 3. The execution time differences are substantial and deviate from the theoretical explanations of time complexity, which may be related to threshold settings. As we did not delve deeply into this, we plan to consider supplementing more experimental data in the future to further explore and more comprehensively evaluate the execution time of the algorithms.

The hardware used in this study includes an HP Z6 G4 Workstation, Intel Xeon Silver 4208 CPU 2.10 GHz (11 M Cache, 2.10 GHz) 8 cores, 64 Gb RAM, SSD HDD, and NVIDIA Quadro RTX4000. OS and coding environments are Win10, MiniConda, Python v3.11 (Panda s1.5.3, Numpy 1.25.0 and Scikit-learn 1.0.2, Basemap modules, etc.).

Table 3. Time complexity and execution time of clustering algorithms.

Algorithm	Time Complexity	Execution Time
HDBSCAN	$O(n \cdot \log n)$	38.37 s
DBSCAN	$O(n^2)$ to $O(n \cdot \log n)$	213.16 s
K-means	$O(n \cdot k \cdot i)$	336.74 s
OPTICS	$O(n^2)$ to $O(n \cdot \log n)$	3091.87 s
Douglas–Peucker	$O(n \cdot \log n)$	-
SQUISH	$O(n)$ to $O(n^2)$	-
MD-Distance	$O(n^2 \cdot p^2)$	-

where n is the number of Data, k is the clusters, i is the iterations.

Evaluating Clustering Result Methods

The performance of algorithms, comparisons with other methods, and evaluations of stability are all crucial. Through performance evaluation, we can effectively measure the accuracy and efficiency of a method, gaining insight into its practical application. Numeric evaluations not only facilitate comparisons between different methods but also highlight the advantages and limitations of each. This evaluation process provides objective indicators of the robustness and stability of the method, which is essential for real-world applications. To further assess our experimental results, we will introduce the Silhouette Score and Davies–Bouldin Index as standard metrics for evaluating the quality of clustering outcomes.

The Silhouette Coefficient is a metric used to evaluate the quality of clustering results. McInnes et al. (2017) [32] utilized the Silhouette Coefficient to assess the quality of clustering. The Silhouette Coefficient is used to measure the compactness and separation of clustering results. This content appears in Section “Evaluating Clustering Result Methods” on page 9 of the paper, where Equation (4) details the calculation method of the Silhouette Coefficient. At the same time, the same section also introduces the Davies–Bouldin Index (DBI) as a metric for evaluating clustering results, including the DBI formula (Equation (5)) and its related description. The theoretical concept is to calculate the average distance of the same clustering data, representing the compactness of that cluster, as well as to compute the core distance between different clusters, representing the separation between different clusters. The values range from -1 to 1 , where a value close to 1 indicates that the data are correctly assigned to their respective cluster, -1 indicates that they have been assigned to the wrong cluster, and 0 indicates that they are on the boundary between two clusters. The formula for the Silhouette Coefficient is as follows in Equation (4):

$$SC = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad -1 \leq S \leq 1 \quad (4)$$

Here, SC is the Silhouette Coefficient for data point i , $a(i)$ is the average distance between data point i and other data points in the same cluster. $b(i)$ is the average core distance between different clusters. The SC allows us to objectively evaluate the rationality of the HDBSCAN clustering results and the clustering performance under different parameter settings.

The Davies–Bouldin Index (DBI) is a metric used to evaluate clustering results, as described by F. Ros et al. (2023) [33]. This index measures the compactness of a cluster using the average distance s_i and evaluates the dissimilarity between different clusters' core distance d_{ij} . When clusters are more compact or the dissimilarity between clusters is greater, the DBI value becomes smaller. A smaller DBI value indicates better clustering performance, signifying that clusters are effectively distinguishable. A DBI value close to 0 implies perfect separation. The calculation formula is as follows, Equation (5):

$$DBI = \frac{1}{k} \sum_{i=1}^k \max(\frac{s_i + s_j}{d_{ij}}) \quad i \neq j \quad (5)$$

where k is the number of clusters, S_i is the average distance from data points within cluster i to its centroid, and d_{ij} is the distance between the centroids of clusters i and j .

5. Experimental Results

Before conducting the clustering analysis, we randomly selected approximately 300 data points to compare the clustering performance of HDBSCAN and DBSCAN (Table 4). During the computation, we recorded the *eps* and *minPts* parameter settings and observed the variations in clustering results. The experimental results indicate that DBSCAN is highly sensitive to changes in the *eps* parameter, requiring multiple manual adjustments to achieve optimal results. In contrast, HDBSCAN converges quickly and produces stable clustering results, demonstrating its parameter robustness and applicability. This study utilized AIS data from the waters off the central Taiwan port, covering a single day with approximately 557,000 records. After filtering, cleaning, and compressing, the dataset was reduced to 293,000 records, comprising 386 trajectory segments, with an average of approximately 756 data points per segment, which served as the reference for minimum clusters.

Table 4. Silhouette Coefficient and Davies–Bouldin Index result.

	HDBSCAN	DBSCAN
<i>SC</i>	0.222	0.142
<i>DBI</i>	1.369	1.433

We selected three-dimensional spatial features of coordinates and heading, as shown in Figure 3. Before optimization, the dataset consisted of X and Y axes representing latitude and longitude coordinates and the Z axis corresponding to heading angles. The figure highlights a yellow vertical region indicating outlier data points in the heading distribution, which could significantly impact clustering calculations. Further analysis revealed that these outliers originated from stationary vessels inside and outside the Taichung port area. To improve clustering performance, we removed data points corresponding to vessels with speeds below 3 knots to eliminate noise from docked or slow-moving ships. Since longitude, latitude, and heading have different scales, we performed normalization of the three-dimensional data, converting them to a standardized range between 0 and 1 to ensure consistency in clustering calculations.

Finally, we applied HDBSCAN’s hierarchical density-based clustering to complete the clustering process. HDBSCAN’s adaptive density thresholding ensured clustering stability while eliminating the need for manual parameter tuning required by DBSCAN, improving clustering accuracy and applicability. The clustering results contained labels $[-1, 0, 1, 2, 3, 4, 5, 6, 7]$, where eight clusters were valid groups and -1 represented noise points. The number of noise points was 37,367, accounting for 12.7% of the total dataset. After clustering, we removed noise points and visualized the results by mapping different clusters with distinct colors in a 3D distribution plot (Figure 4) and a 2D trajectory cluster map (Figure 5).

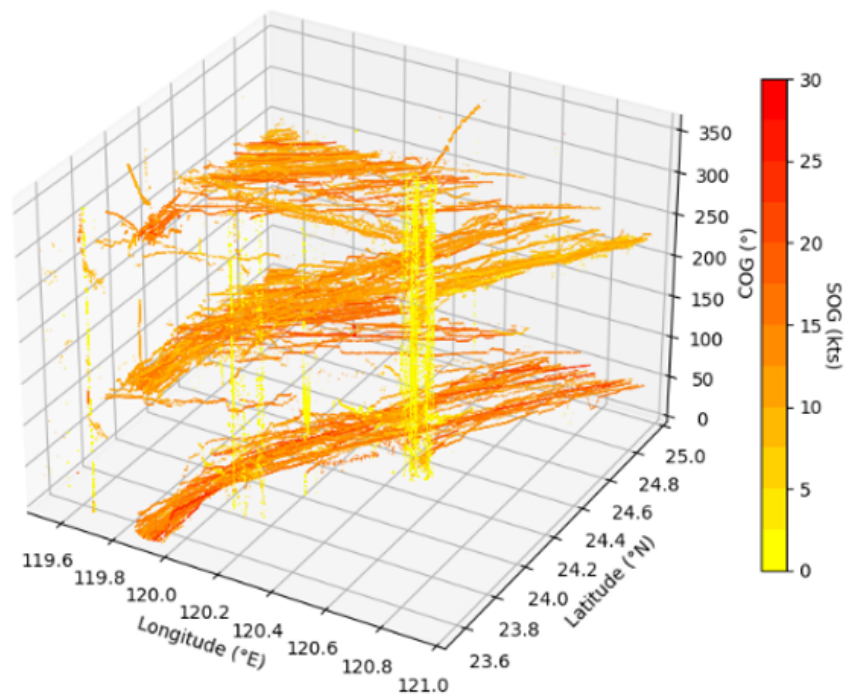


Figure 3. The low-speed data points distributed vertically at specific coordinates.

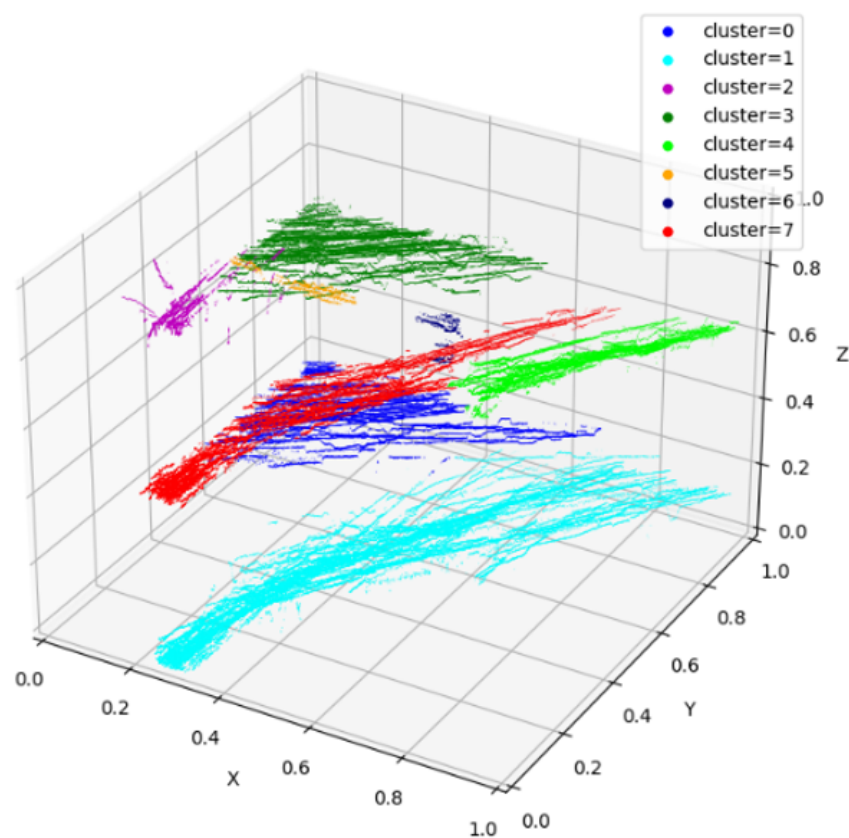


Figure 4. Different colors clearly represent distinct HDBSCAN clustering results.

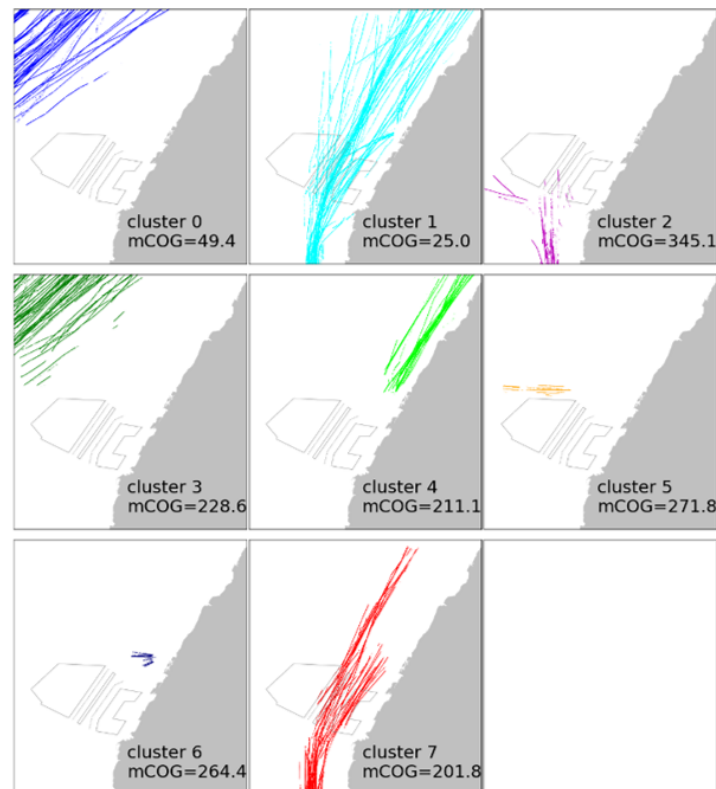


Figure 5. Each cluster distribution.

Through visual inspection, we could clearly observe navigation patterns and cluster distributions, such as high-density maritime areas and fixed routes. Beyond visual inspection, we further assessed the clustering results using two evaluation metrics—the Silhouette Coefficient (SC) and the Davies–Bouldin Index (DBI).

- The DBI measures the compactness within clusters and the separation between clusters, where lower values indicate better clustering quality.
- SC evaluates the similarity of a single data point to its own cluster relative to the nearest neighboring cluster. SC values range from -1 to 1 , with higher values indicating that data points are closer to their own cluster center and further from other clusters, signifying better clustering performance.

As shown in Table 3, HDBSCAN achieved a DBI of 1.369 (lower than DBSCAN's 1.433) and an SC of 0.222 (higher than DBSCAN's 0.142). This indicates that HDBSCAN exhibits better cluster compactness and overall superior clustering performance. Although the differences in evaluation metrics are not substantial, HDBSCAN slightly outperforms DBSCAN in trajectory clustering.

The results of this study demonstrate that HDBSCAN performs exceptionally well in big data clustering analysis. By leveraging the computational power of clustering algorithms, we can rapidly classify ship trajectories for maritime management, which is crucial for overseeing hazardous or suspicious vessel movements. The contributions of this study not only validate the practical effectiveness of HDBSCAN but also scientifically verify the quality of clustering results through evaluation metrics. Furthermore, it provides a reliable technical foundation for intelligent maritime management, offering significant implications for the future development of maritime intelligent transportation and navigational safety systems.

6. Conclusions

This study aims to utilize the HDBSCAN algorithm for clustering analysis of vessel trajectory big data to identify different shipping patterns and behaviors. By analyzing 293,000 historical Automatic Identification System (AIS) data points, we employed various preprocessing techniques, including filtering, compression, and cleaning. Notably, the removal of low-speed data outside ports significantly enhanced the accuracy of the clustering results. We normalized spatial features such as coordinates and heading angles, demonstrating the clear advantages of HDBSCAN in hierarchically processing data of varying densities. This approach successfully identified multiple effective trajectory clusters and provided a visual exploration of the geographical distribution and heading characteristics of these clusters. Additionally, We also listed the O notation time complexities of other methods and calculated the execution times of several methods, demonstrating that the execution performance of HDBSCAN outperforms the other methods and validated the clustering results using evaluation metrics like SC (Silhouette Coefficient) and DBI (Davies–Bouldin Index). The results of the trajectory clustering algorithm can provide an important reference for navigation safety and path optimization. In the future, we plan to expand the time and scope of data collection if conditions permit, to collect and integrate more data from different periods, such as the comparison of data changes before and after COVID, or to analyze different regions or combine other data such as ocean currents and weather to further prove the universality of this method and enable the technology to be extended to maritime regulatory applications.

Author Contributions: Conceptualization: W.-R.Y. and S.-M.W.; Software: W.-R.Y. and Q.-Y.Z.; Formal analysis: W.-H.L. and Q.-Y.Z.; Investigation: and M.-Y.T.; Q.-Y.Z.; Resources: W.-R.Y. and S.-M.W.; Writing—original draft: W.-R.Y. and J.-C.C.; Writing—review and editing: T.-J.S. and S.-M.W.; Supervision: T.-J.S., J.-C.C. and S.-M.W.; Project administration: W.-R.Y., T.-J.S. and S.-M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported financially in part by the National Science and Technology Council, ROC (No. 113-2221-E-230-004).

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki and approved by the Institutional Review Board (or Ethics Committee) of Yuan’s General Hospital (protocol code: 20220531 approval) for studies involving humans.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *kdd* **1996**, *96*, 226–231.
2. Campello, R.J.; Moulavi, D.; Sander, J. Density-based clustering based on hierarchical density estimates. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Gold Coast, Australia, 14–17 April 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 160–172.
3. Yuan, G.; Xia, S.; Zhang, L.; Zhou, Y.; Ji, C. Trajectory outlier detection algorithm based on structural features. *J. Comput. Inf. Syst.* **2011**, *7*, 4137–4144.
4. Lin, K.; Xu, Z.; Qiu, M.; Wang, X.; Han, T. Noise filtering, trajectory compression and trajectory segmentation on GPS data. In Proceedings of the 2016 11th International Conference on Computer Science & Education (ICCSE), Nagoya, Japan, 23–25 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 490–495.
5. Early, J.J.; Sykulski, A.M. Smoothing and interpolating noisy GPS data with smoothing splines. *J. Atmos. Ocean. Technol.* **2020**, *37*, 449–465. [[CrossRef](#)]

6. Laxhammar, R. Anomaly detection for sea surveillance. In Proceedings of the 2008 11th international conference on information fusion, Cologne, Germany, 30 June–3 July 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–8.
7. Douglas, D.H.; Peucker, T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartogr. Int. J. Geogr. Inf. Geovisualization* **1973**, *10*, 112–122. [\[CrossRef\]](#)
8. Zhu, F.X.; Miao, L.M.; Liu, W. Research on vessel trajectory multi-dimensional compression algorithm based on Douglas-Peucker theory. *Appl. Mech. Mater.* **2014**, *694*, 59–62. [\[CrossRef\]](#)
9. Zhao, L.; Shi, G. A method for simplifying ship trajectory based on improved Douglas–Peucker algorithm. *Ocean Eng.* **2018**, *166*, 37–46. [\[CrossRef\]](#)
10. Zhang, S.K.; Liu, Z.J.; Cai, Y.; Wu, Z.L.; Shi, G.Y. AIS trajectories simplification and threshold determination. *J. Navig.* **2016**, *69*, 729–744. [\[CrossRef\]](#)
11. Zhao, L.; Shi, G. A trajectory clustering method based on Douglas-Peucker compression and density for marine traffic pattern recognition. *Ocean Eng.* **2019**, *172*, 456–467. [\[CrossRef\]](#)
12. Cao, W.; Li, Y. DOTS: An online and near-optimal trajectory simplification algorithm. *J. Syst. Softw.* **2017**, *126*, 34–44. [\[CrossRef\]](#)
13. Meng, Q.; Yu, X.; Yao, C.; Li, X.; Li, P.; Zhao, X. Improvement of OPW-TR algorithm for compressing GPS trajectory data. *J. Inf. Process. Syst.* **2017**, *13*, 533–545.
14. Muckell, J.; Hwang, J.H.; Lawson, C.T.; Ravi, S. Algorithms for compressing GPS trajectory data: An empirical evaluation. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 402–405.
15. Muckell, J.; Hwang, J.H.; Patil, V.; Lawson, C.T.; Ping, F.; Ravi, S. SQUISH: An online approach for GPS trajectory compression. In Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications, Washington, DC, USA, 23–25 May 2011; pp. 1–8.
16. Chen, Y.; Zhou, L.; Bouguila, N.; Wang, C.; Chen, Y.; Du, J. BLOCK-DBSCAN: Fast clustering for large scale data. *Pattern Recognit.* **2021**, *109*, 107624. [\[CrossRef\]](#)
17. Elbatta, M.T.; Ashour, W.M. A dynamic method for discovering density varied clusters. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2013**, *6*, 123–134.
18. Hou, J.; Gao, H.; Li, X. DSets-DBSCAN: A parameter-free clustering algorithm. *IEEE Trans. Image Process.* **2016**, *25*, 3182–3193. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Smiti, A.; Eloudi, Z. Soft DBSCAN: Improving DBSCAN clustering method using fuzzy set theory. In Proceedings of the 2013 6th International Conference on Human System Interactions (HSI), Sopot, Poland, 6–8 June 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 380–385.
20. Chen, X.; Min, Y.; Zhao, Y.; Wang, P. GMDSCAN: Multi-density DBSCAN cluster based on grid. In Proceedings of the 2008 IEEE International Conference on e-Business Engineering, Xi'an, China, 22–24 October 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 780–783.
21. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
22. Deng, Z.; Hu, Y.; Zhu, M.; Huang, X.; Du, B. A scalable and fast OPTICS for clustering trajectory big data. *Clust. Comput.* **2015**, *18*, 549–562. [\[CrossRef\]](#)
23. Jiang, H.; Li, J.; Yi, S.; Wang, X.; Hu, X. A new hybrid method based on partitioning-based DBSCAN and ant clustering. *Expert Syst. Appl.* **2011**, *38*, 9373–9381. [\[CrossRef\]](#)
24. Götz, M.; Bodenstein, C.; Riedel, M. HPDBSCAN: Highly parallel DBSCAN. In Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, Austin, TX, USA, 15–20 November 2015; pp. 1–10.
25. Ienco, D.; Bordogna, G. Fuzzy extensions of the DBScan clustering algorithm. *Soft Comput.* **2018**, *22*, 1719–1730. [\[CrossRef\]](#)
26. Trisminingsih, R.; Shaztika, S.S. ST-DBSCAN clustering module in SpagoBI for hotspots distribution in Indonesia. In Proceedings of the 2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, Indonesia, 19–20 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 327–330.
27. Xie, Y.; Jia, X.; Shekhar, S.; Bao, H.; Zhou, X. Significant DBSCAN+: Statistically robust density-based clustering. *ACM Trans. Intell. Syst. Technol. (TIST)* **2021**, *12*, 1–26. [\[CrossRef\]](#)
28. Gholizadeh, N.; Saadatfar, H.; Hanafi, N. K-DBSCAN: An improved DBSCAN algorithm for big data. *J. Supercomput.* **2021**, *77*, 6214–6235. [\[CrossRef\]](#)
29. ITU-T. *Traffic Control and Congestion Control in B-ISDN*; Recommendation I.371; International Telecommunication Union: Geneva, Switzerland, 1996.
30. Campello, R.J.; Moulavi, D.; Zimek, A.; Sander, J. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Trans. Knowl. Discov. Data (TKDD)* **2015**, *10*, 1–51. [\[CrossRef\]](#)
31. Sedgewick, R.; Wayne, K. *Algorithms*; Addison-Wesley Professional: Boston, MA, USA, 2011.

32. McInnes, L.; Healy, J.; Astels, S. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* **2017**, *2*, 205. [[CrossRef](#)]
33. Ros, F.; Riad, R.; Guillaume, S. PDBI: A partitioning Davies-Bouldin index for clustering evaluation. *J. Open Source Softw.* **2023**, *5282*, 178–199. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.