symmetry

MDPI

Review

A Systematic Literature Review on Identifying Patterns Using Unsupervised Clustering Algorithms: A Data Mining Perspective

Mahnoor Chaudhry [1], Imran Shafi [1], Mahnoor Mahnoor [1], Debora Libertad Ramírez Vargas [2,3,4], Ernesto Bautista Thompson [2,3,5] and Imran Ashraf [6,*]

[1] College of Electrical and Mechanical Engineering, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan; mchaudhary.cs22ceme@student.nust.edu.pk (M.C.); imranshafi@ceme.nust.edu.pk (I.S.); mmahnoor.cs22ceme@student.nust.edu.pk (M.M.)
[2] Higher Polytechnic School, Universidad Europea del Atlántico, Isabel Torres 21, 39011 Santander, Spain; debora.ramirez@unini.edu.mx (D.L.R.V.); ernesto.bautista@unini.edu.mx (E.B.T.)
[3] Department of Project Management, Universidad Internacional Iberoamericana, Campeche 24560, Mexico
[4] Universidad de La Romana, La Romana, Dominican Republic
[5] Universidade Internacional do Cuanza, Cuito EN250, Bié, Angola
[6] Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Republic of Korea
* Correspondence: imranashraf@ynu.ac.kr

Abstract: Data mining is an analytical approach that contributes to achieving a solution to many problems by extracting previously unknown, fascinating, nontrivial, and potentially valuable information from massive datasets. Clustering in data mining is used for splitting or segmenting data items/points into meaningful groups and clusters by grouping the items that are near to each other based on certain statistics. This paper covers various elements of clustering, such as algorithmic methodologies, applications, clustering assessment measurement, and researcher-proposed enhancements with their impact on data mining thorough grasp of clustering algorithms, its applications, and the advances achieved in the existing literature. This study includes a literature search for papers published between 1995 and 2023, including conference and journal publications. The study begins by outlining fundamental clustering techniques along with algorithm improvements and emphasizing their advantages and limitations in comparison to other clustering algorithms. It investigates the evolution measures for clustering algorithms with an emphasis on metrics used to gauge clustering quality, such as the F-measure and the Rand Index. This study includes a variety of clustering-related topics, such as algorithmic approaches, practical applications, metrics for clustering evaluation, and researcher-proposed improvements. It addresses numerous methodologies offered to increase the convergence speed, resilience, and accuracy of clustering, such as initialization procedures, distance measures, and optimization strategies. The work concludes by emphasizing clustering as an active research area driven by the need to identify significant patterns and structures in data, enhance knowledge acquisition, and improve decision making across different domains. This study aims to contribute to the broader knowledge base of data mining practitioners and researchers, facilitating informed decision making and fostering advancements in the field through a thorough analysis of algorithmic enhancements, clustering assessment metrics, and optimization strategies.

Keywords: clustering; distance measures; data mining; evolution measures; symmetry

Citation: Chaudhry, M.; Shafi, I.; Mahnoor, M.; Vargas, D.L.R.; Thompson, E.B.; Ashraf, I. A Systematic Literature Review on Identifying Patterns Using Unsupervised Clustering Algorithms: A Data Mining Perspective. *Symmetry* **2023**, *15*, 1679. https://doi.org/10.3390/sym15091679

Academic Editor: Shuang Xu

Received: 23 July 2023
Revised: 23 August 2023
Accepted: 29 August 2023
Published: 31 August 2023

1. Introduction

In today's globalized world, organizations are confronted with an explosive proliferation of data from many sources, making it difficult to focus on important information. Artificial intelligence (AI) is developing as a pillar of contemporary problem solving, leading to a new era of innovation as a result of technological advancements. The extraordinary

advancements in a variety of sectors have been made possible by AI's capacity to analyze enormous volumes of data and uncover hidden insights. An essential component of AI called data mining is used to extract useful information from large databases. Data mining is an analytical approach that contributes to achieving a solution to this problem by extracting previously unknown, fascinating, nontrivial, and potentially valuable information from massive datasets. According to Shukor [1], data mining is the act of discovering patterns, linkages, changes, variations, and distinctive structures in data preserved from diverse sources. To extract information and insights from data, data scientists employ a wide range of expensive computer approaches. In a wide range of industries, including business, healthcare, and finance, data mining has critical applications. Data mining aids in decision making, marketing strategy optimization, and consumer behavior analysis in business. Data mining in healthcare helps with illness diagnosis, patient outcome prediction, and treatment plan optimization [2]. Data mining algorithms provide a variety of patterns or insights to be found across the data mining processes. The most common of these capabilities are summarization, characterization, discrimination, association, clustering, classification, outlier analysis, regression modeling, and pattern analysis [3]. This powerful tool investigates data utilizing a range of methods, like association rule mining, clustering, classification, and anomaly detection. Data mining is employed in a variety of areas like financial services, marketing, medical treatment, and social networking development [4–8]. It is an essential tool for firms that wish to make decisions based on data-driven insights.

Data mining encompasses a variety of methodologies from several domains [9] including data analysis, database platforms, artificial intelligence techniques, recognition of patterns, visualization, data retrieval, and computational performance, whereas statistics, database technology, and artificial intelligence are the primary sources of data mining innovations. By extracting important information from enormous datasets, these technologies enable organizations to make logical decisions based on insights generated by data. Data mining algorithms provide a variety of patterns and insights across the data mining processes. The most common of these capabilities are summarization, characterization, discrimination, association, clustering, classification, outlier analysis, regression modeling, and pattern analysis [10]. These features help the extraction of relevant information from datasets by recognizing associations, patterns, and anomalies, along with modeling anticipated trends and behaviors.

Researchers have looked into a range of methodologies that are utilized in data mining. Among these techniques are AI, statistics, artificial neural networks, database and data storage structures, algorithms based on genetics, fuzzy sets, visualization, and others [11]. These techniques are used to retrieve relevant information from big datasets by discovering patterns, grouping, and anomalies, and by modeling forthcoming trends and behavior. Scholars are continually studying novel ways and methods to improve the precision and efficacy of data mining jobs and to meet the increasing expectations of organizations in today's data-driven environment. Many researchers have created numerous algorithms, also known as techniques, for carrying out data mining activities that utilize data mining techniques. Examples include the Apriori method, Naive Bayesian, rule-based classification, k-nearest neighbor, k-Means, k-medoid, partition around medoids (PAM), clustering large applications (CLARA), CLIQUE, clustering large applications based upon randomized search (CLARANS), statistical information grid (STING), and others [12].

Data mining may be applied in a variety of fields, for example, time-lapse data mining, mining the web, temporal information mining, spatial information mining, temporal–spatial data mining, educational information mining, commerce, healthcare, sciences, technical data mining, and so on. Each domain may have a few different data mining applications [13]. It is a collection of application domains in which a variety of data mining functions can be applied. Statistical analysis of data, market-basket analysis, detection of intrusions, identification of fraud, recommendation systems, cancer diagnosis, and other applications can be used [14].

Data mining is a method of identifying patterns and insights in massive databases using computational and statistical approaches. It entails collecting useful information from massive volumes of data and applying it to make sound judgments. Clustering, classification, association rule mining, regression analysis, and outlier identification are all data mining approaches. However, the three basic data mining approaches are association rule mining, classification, and clustering. Association rule mining produces implications between multiple data sets; classification is a form of supervised learning that categorizes data items [15], and regression-analysis is employed to imitate the relationship between a dependent variable and numerous independent variables [16]. Outlier detection includes recognizing data points that stray considerably from the norm, while clustering is a form of unsupervised learning that organizes data items into unknown subgroups [17].

Clustering, which includes putting data points into useful clusters based on underlying patterns, is one of the key techniques used in data mining. This method has shown to be quite useful in a variety of fields and makes it easier to comprehend large datasets. Clustering plays an increasingly prominent role as datasets continue to expand in size and complexity. A crucial tool for discovering hidden patterns without the need for explicit labeling is unsupervised clustering algorithms, a subset of AI approaches [10]. These algorithms work by grouping data points into clusters based on shared characteristics, aiding in the clarification of significant connections and insights. This powerful tool investigates data utilizing a range of methods, like association rule mining, clustering, classification, and anomaly detection. Data mining is employed in a variety of areas, like financial services, marketing, medical treatment, and social networking development. It is an essential tool for firms that wish to make decisions based on data-driven insights [18]. Figure 1 depicts the data mining strategies which are used in the existing literature. Clustering is one of the widely used approaches to data mining and holds significant importance. This study considers both supervised and unsupervised clustering approaches from a data mining perspective and provides a comprehensive review of the clustering approaches.

Figure 1. Categorization of data mining techniques.

The main goal of this study is to thoroughly examine unsupervised clustering techniques in the context of data mining. This study will examine the effectiveness and efficiency of clustering, the application of algorithmic techniques across a range of domains, and the effects of suggested improvements. This study's sub-objectives include assessing clustering assessment measures, looking at optimization techniques, and investigating

how algorithms change over time. The basic assumption is that a thorough analysis of unsupervised clustering algorithms would reveal their various strengths, weaknesses, and potential for knowledge discovery, advancing data mining techniques. This study analyzes the benefits and drawbacks of various unsupervised clustering methods, identifies practical uses for them, and describes recent developments in the field. The study aims to give practitioners and scholars a comprehensive grasp of the current approaches by addressing these factors, opening the way for improved information extraction and data analysis.

This review is further divided into five sections. The background of the clustering approach is presented in Section 2. Section 3 provides the adopted review methodology including research questions, paper selection, and inclusion and exclusion criteria. It is followed by a discussion of the findings related to set research questions in Section 4. Section 5 provides a brief discussion of the findings. Conclusions and future directions are given in Section 6.

2. Background

Clustering is a key data mining approach for splitting databases into subgroups so that visualization and efficient extraction of valuable data from enormous amounts of structured and unstructured data points can be realized. In fact, clustering is a rational arrangement of large amounts of unstructured data in order to analyze it [19]. Meanwhile, clustering is regarded as a difficult issue in the field of data mining, and it has gained significant attention among scholars in the past. Furthermore, this displays the applicability of data points for each technique and identifies numerous parametric variables that may be used to compare various clustering algorithms. These characteristics are the data size, managing noisy information, dataset category, cluster form, input parameter and complexities, database applicability, overlapping cluster, rapidity, the effect of input control, and scalability.

Clustering is a technique for splitting/segmenting data items (or data points) into groups and clusters. Items that are near to each other are grouped together. Clustering, similar to classification, identifies related data items; however, unlike classification, the class identities are unknown (like unsupervised learning) [11]. Cluster analysis is a common approach that is utilized not just in the field of data mining but also in statistics, segmentation of pictures, recognition of patterns, object recognition, retrieving data, computational biology, and other fields [19–22].

The origin of clustering may be linked back to the initial stages of statistics and pattern detection [23]. In the beginning, clustering algorithms concentrated on dividing the dataset into distinct sections with the goal of maximizing similarity inside each group while minimizing dissimilarity across groups. These approaches, like K-means and K-medoids, served as the foundation for many future advances in clustering. Scholars gradually realized that the premise of distinct clusters could not be true for all forms of data. This resulted in the development of algorithms capable of dealing with clusters that overlap and clusters with complicated forms. Hierarchical clustering methods were created to portray clustering outcomes within a hierarchical framework, enabling study at many granularity stages [24]. Clustering approaches have been developed even further as a result of developments in artificial intelligent computational methodologies, and the accessibility of huge datasets. To handle various types of data and clustering events, new methods like density-based clustering, for example, DBSCAN, model-based clustering, for example, Gaussian mixture models, and grid-based clustering, have recently been developed. Evaluation metrics have proven crucial in the formation of techniques for clustering. To determine the effectiveness of clustering findings, many metrics like clustering efficiency, silhouette coefficient, and Rand index have been proposed [25]. These criteria aid in the comparison and selection of the best clustering method for a particular task and dataset.

Clustering techniques have encountered additional hurdles since the emergence of big data and the explosion of datasets with high dimensions. Kang et al. [26] introduced subspace clustering and ensemble clustering techniques that were originally established for handling data with many sub-spaces and to integrate the results of various clustering

methods. Efforts have also been made to increase the scalability and effectiveness of techniques for clustering when dealing with huge datasets. Clustering has an extended history of algorithmic advancement, assessment approaches, and adaptability to changing data properties and processing constraints. Clustering is a thriving research topic, driven by the desire to detect significant patterns and frameworks in data, promote the acquisition of knowledge, and enhance methods for making decisions across several domains [27].

Chitra et al. discuss various forms of clustering in [28]. A cluster constitutes one of the data mining procedures, according to them. It is a form of unsupervised learning that involves grouping a bunch of identical items into a single cluster. The co-authors of this work compare several methods of clustering such as partition-based, hierarchical, grid-based, and density-based clustering. There are several clustering approaches in the literature, including partitioning, hierarchical, model-based, density-based, and grid-based clustering, among others. Figure 2 depicts the data mining hierarchy as well as the approaches and their types.

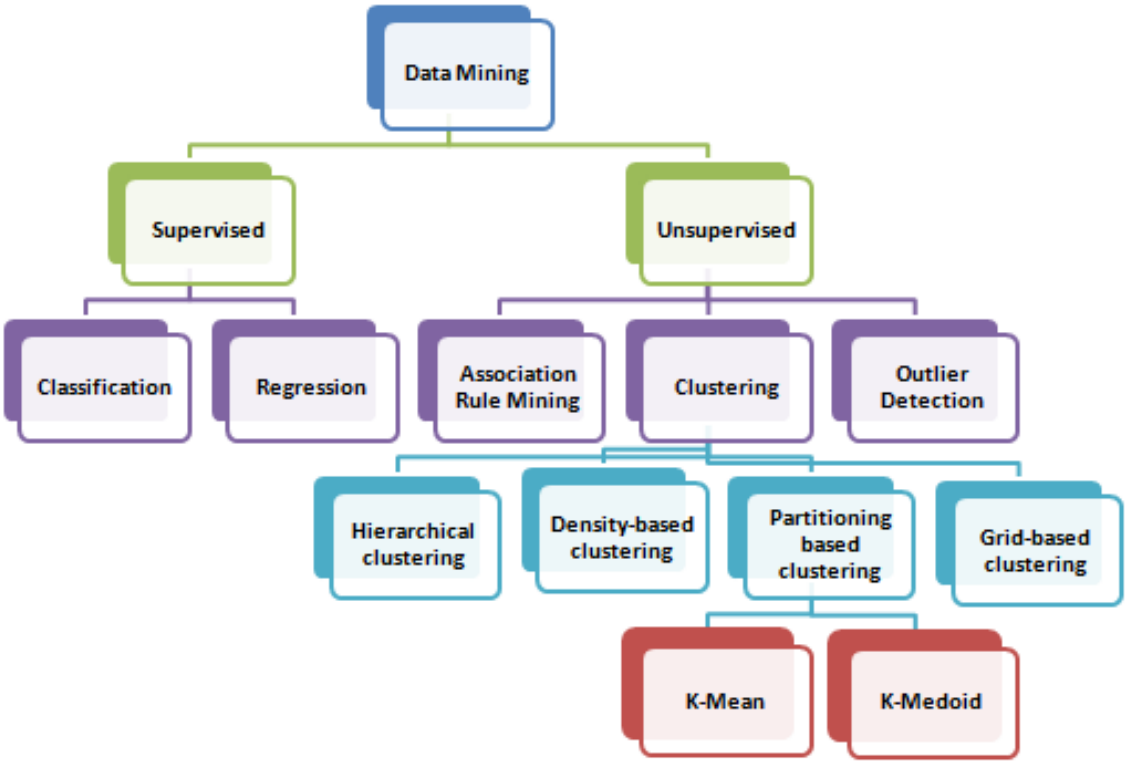

Figure 2. Categorization of clustering techniques for data mining.

3. Methodology

This section defines the article selection criteria, data sources, and the search strategy for data extraction and analysis procedure. Figure 3 shows the workflow of the methodology adopted for this study.

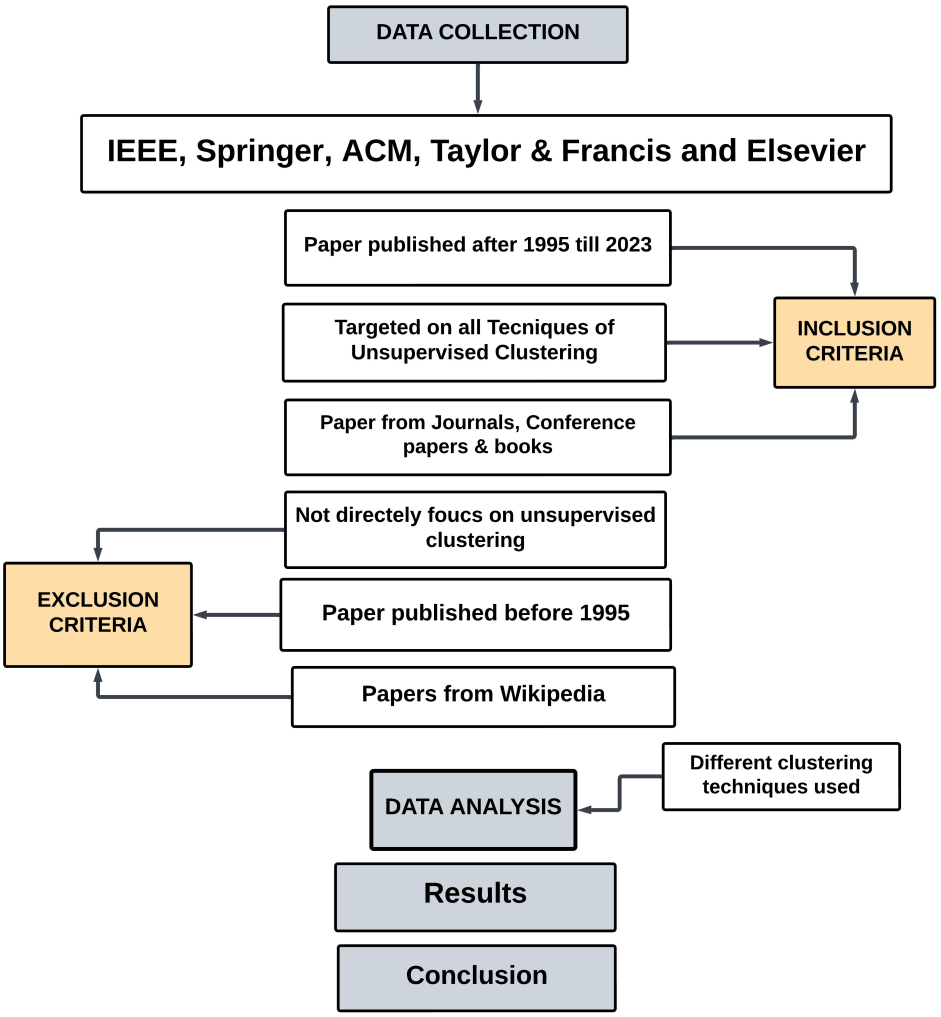

Figure 3. Methodology adopted for the literature review.

3.1. Research Questions

The following are the research questions formulated to analyze the relevant studies:

i. What approaches and algorithms are currently available in clustering?
ii. What are the benefits and drawbacks of various clustering techniques?
iii. What are the clustering evaluation measures to consider when selecting a centroid finding method?
iv. What are the applications or fields where some clustering algorithms outperform others?

3.2. Inclusion and Exclusion Criteria

The most significant aspect of a systematic literature review (SLR) is the inclusion and exclusion criteria. This is used to choose or reject research articles. This study proposes the following six criteria for study inclusion and exclusion.

3.2.1. Subject

A key point in this SLR is unsupervised clustering algorithms. As a result, the research papers should include only the role of data mining in unsupervised clustering algorithms for looking at optimization techniques, and investigating how algorithms change over time.

3.2.2. Application Research

The selected research has a crucial positive effect on the effectiveness and efficiency of clustering, also taking into account the use of algorithmic approaches across various domains and the effects of suggested improvements, by closely examining clustering assessment metrics, exploring optimization techniques, and following the development of algorithms over time.

3.2.3. Publication Year

Recent research is predicated on the background of preceding studies. As a result, we choose a well-balanced publishing year duration from 2003 to 2022 for this SLR. This timeline not only includes the recent clustering techniques but also systematically includes past achievements.

3.2.4. Publisher

This SLR considers six major scientific sources for the collection of research papers, which are as follows:

- Google scholar;
- IEEE;
- Springer;
- ACM;
- Taylor & Francis;
- Elsevier.

3.2.5. Validation of Proposal

For the best possible study, the idea must be properly validated. As a result, the research should only be chosen if the idea is validated using the best clustering technique. There are research works in this domain where inadequate information is supplied for idea validation. The SLR discards those research papers which have inadequate or missing information.

3.2.6. Repetition

In this SLR, papers with almost identical study materials are removed and only the most reliable and consistent data are considered. We carried out this SLR using the previously given inclusion and exclusion criteria. The research is chosen specifically if all inclusion and exclusion criteria are satisfied. Even if only one inclusion and exclusion criterion is ignored, the research is dismissed.

3.3. Risks of Bias

3.3.1. Methods for Risk of Bias Assessment

Reviewers independently evaluated the risk of bias for the included studies. Each study's risk of bias was assessed using a standardized tool, which improved the assessment's consistency and impartiality.

3.3.2. Reviewer Independence

To reduce bias and increase the reliability of the evaluation, the reviewers each worked separately during the risk of bias assessment process.

3.3.3. Automated Tools in Bias Assessment

Where appropriate, automation techniques were used to speed up the risk of the bias assessment process and guarantee that it was carried out in a methodical manner.

3.3.4. Impact Measures

Specific impact measures were chosen for the synthesis and presentation of results for each outcome mentioned in the relevant research papers. These metrics were chosen in light of their applicability and relevance to the research topic.

3.4. Search Process

To obtain the most significant results, we carried out automated searches. The searching method was carried out in electronic databases and validated by data analysts [12]. The selected sources were chosen because they contain high-quality networking articles and conference papers. The search period is between 2002 and 2023. The search query was created by analyzing the following search terms in order to obtain the necessary information from the selected sources. Keywords include 'data mining', 'clustering', 'partition based clustering', 'k-mean', 'k-medoid', 'density based clustering', 'hierarchical clustering', and so on. Table 1 shows the search process used for various databases and results from each database against searched terms.

Table 1. The search process uses various search terms to filter the results.

Sr. no.	Search Term	IEEE	Springer	ACM
1	Data mining	32	14	09
2	Clustering	34	23	27
3	Different clustering techniques	28	18	13
4	Partition based clustering	74	33	10
5	Hierarchal Clustering	82	57	18
6	Density-based clustering	92	69	16

A thorough search of numerous databases produced an initial pool of 698 studies for the systematic review. A total of 100 duplicate studies were successfully deleted after careful efforts to do so; this left a well-curated group of 598 studies for future analysis. These 598 studies underwent a stringent screening procedure, which involved a careful examination of their titles and abstracts. After the screening step, 300 full-text publications were obtained and evaluated to see if they qualified for the study. Following a rigorous evaluation of these full-text papers, 298 research papers that did not meet the predetermined inclusion criteria were excluded. The selection procedure was streamlined by the strict application of exclusion criteria, which led to the final inclusion of 143 research papers.

3.5. Quality Assessment

In quality assessment criteria, we analyzed all the collected studies based on the coherence and relevancy of addressing the research questions defined. Figure 4 shows the distribution of papers concerning the venue, including conferences, journals, and books.

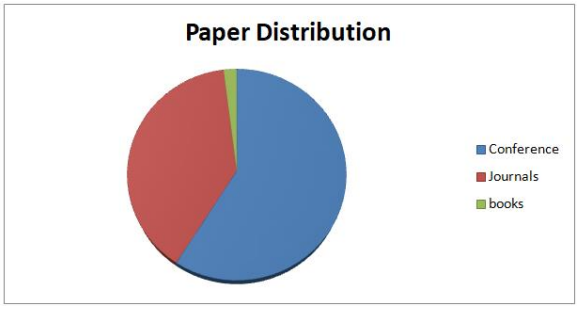

Figure 4. Distribution of selected papers.

Figure 5 shows the primary sources of references. It shows that IEEE has the largest number of references at 54, Elsevier has 39 references, Springer has 19 references, ACM has 4 references, and the remaining 27 references are from other categories, which include different sources like ResearchGate, etc.

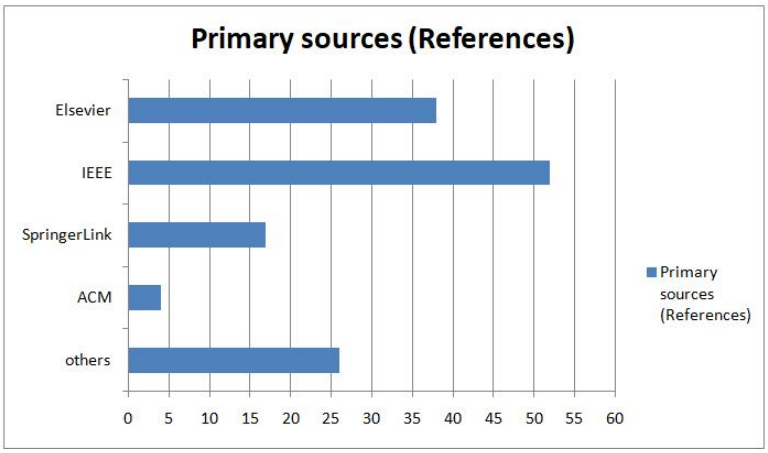

Figure 5. Distribution of selected papers concerning sources.

3.6. Data Extraction and Synthesis

After selecting research based on inclusion and exclusion criteria, the data extraction and synthesis procedure is carried out. Initially, the key components are taken from chosen research. Table 2 shows the details of data extraction and synthesis.

Table 2. Elements of data extraction and synthesis.

Sr. No.	Extracted Elements	Particulars
1	Bibliographic information	Title, author name, year to publications, publisher details, and type of research (i.e., journal and conference papers or books)
2	Abstract	Proposal of research paper
3	Limitation	Selected research limitation in accessing the goals.
4	Validation method	Validation method is used in each selected research.
	Data extraction with synthesis	
5	Clustering techniques	Leading clustering techniques. The result is summarized in Sections 4.1–4.4.
6	Comparison	Leading clustering techniques in terms of accuracy, complexities, and with their limitations. The results are summarized in Section 4.5.
7	Hyperparameter Tuning in Clustering Algorithms Approaches	Hyperparameter Tuning for Clustering is summarized in Section 4.6.
8	Different evolution measures for clustering techniques	Different Evolution Measures for clustering techniques are utilized in selected studies. And the result is summarized in Section 4.7.
9	Application of clustering	Applications of different clustering are utilized in selected studies. And the results are summarized in Section 4.7 and the last figure and the last table.

4. Results

This section provides precise results in order to offer authentic answers to research questions. To acquire pertinent and important findings, the literature review that was

conducted for this study followed a strict and organized methodology. High-quality networking articles and conference papers from the years 1995 to 2023 were chosen using automated searches in electronic databases.

Important terms related to data mining and clustering techniques were included in the search criteria. Following this procedure, 58 eligible studies (partition clustering $n = 19$, hierarchical clustering $n = 13$, other clustering $n = 14$, clustering application, and evolution measures $n = 12$) are included in this review. A total of 58 papers were considered, and they were divided into partition clustering, hierarchical clustering, other clustering approaches, clustering application, and evolution measures of clustering. The coherence and applicability of the gathered research were assessed as part of the quality evaluation procedure. A noteworthy result of the inclusion criteria was a varied mix of study forms.

Another crucial element for determining the quality of SLR is the form of research that has been chosen, such as a journal, conference paper, or book. Although we attempted to choose as many conference papers as feasible, we were successful in finding 32 conference papers (out of 58) that were totally compatible with the inclusion and exclusion criteria, whereas 39% of selected research is from journals, 59% from conferences, and 2% from books. Figure 6 shows the distribution of the research papers.

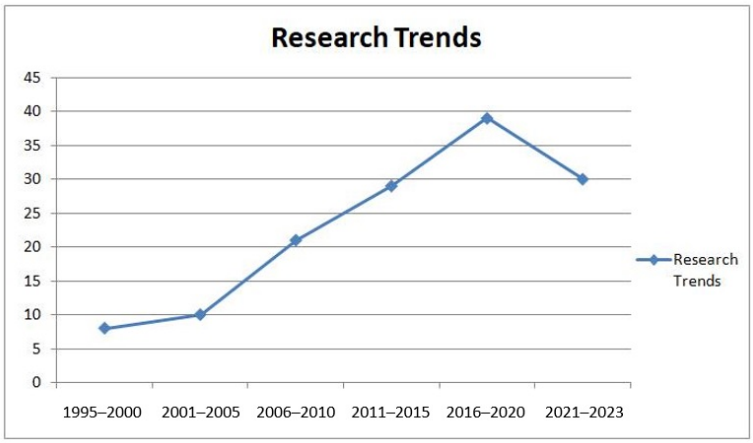

Figure 6. Distribution of papers by year.

From the selected studies, different clustering techniques including partitioning, hierarchical, density-based, and grid-based clustering used by experts are identified and included in this section. Finally, the applications of clustering are provided. Figure 7 shows the hierarchy of the clustering approaches which are covered in this paper.

4.1. Hierarchical Clustering

Hierarchical clustering is a popular unsupervised learning strategy for grouping similar data elements. It creates a cluster of hierarchical structure by repeatedly merging and dividing clusters based on similarity and dissimilarity [29]. The core concept underlying hierarchical clustering is to construct a dendrogram, which is a tree-like framework that depicts the connections between data points and clusters. The dendrogram begins with every data point as a separate cluster and eventually combines related groups according to a similarity and distance value. The algorithm assesses the closeness of clusters and picks the clusters to combine at each stage [30]. According to the data and problematic domain, the similarity and distance metrics employed in hierarchical clustering might differ. Distance measures that are commonly employed comprise Euclidean distance, Manhattan distance, and correlation distance. These measurements assess the dissimilarity and similarity of data points and serve as a basis for the clustering method [31]. In the end, hierarchical clustering is a strong and adaptable clustering algorithm that provides a hierarchical visualization of data. It provides observations into the dataset's linkages and patterns and enables

study at various degrees of depth. There are a few algorithms that employ hierarchical clustering in various ways: agglomerative and divisive hierarchical clustering, balanced iterative reducing and clustering with hierarchies (BIRCH), clustering using representatives (CURE), robust clustering using links (ROCK), and clustering heterogeneous and attributed multi-modal environments using localized similarity indicators (CHAMELEON).

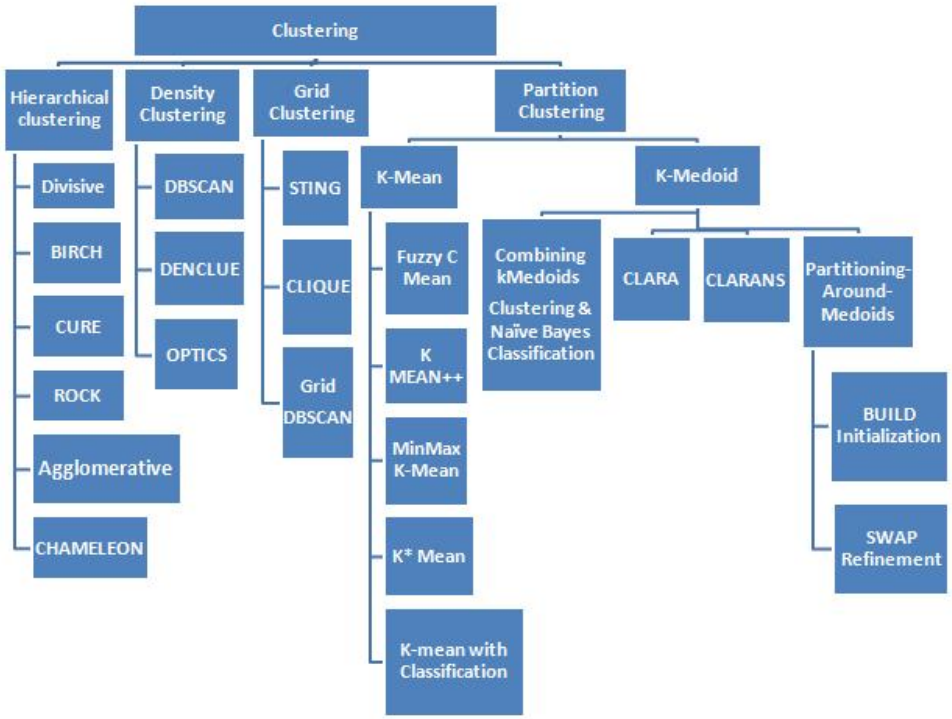

Figure 7. Hierarchy of the clustering approaches.

4.1.1. Divisive Method

Divisive clustering is the inverse of agglomerative clustering. This technique is introduced by M. E. Celebi, Q. Wen, and S. Hwang. In this technique, all of the data items are initially joined into one cluster and then subsequently split into small sub-clusters until a stopping condition is fulfilled. Using this method, a top-down hierarchy is formed [32]. To segment the data depending on particular criteria, it employs a "divide & conquer" technique. Several steps are involved in this process, and it begins with an individual cluster that contains all data items. It identifies the best strategy to separate the cluster using clustering criteria or dissimilarity measurement. Depending on the specified criteria, the cluster is broken into a minimum of two sub-clusters. These steps are continued for every sub-cluster until there is a single data point in every sub-cluster. The binary tree, referred to as a dendrogram, may be used to describe the consequent hierarchical structure [33]. This technique is computationally demanding, particularly for big datasets, because it involves separating the clusters iteratively. However, it offers a thorough hierarchical framework that identifies the connections between data elements at various degrees of granularity.

4.1.2. Agglomerative Method

S. S. Negi and M. K. Jindal published one of the first papers on agglomerative clustering in 1979. Agglomerative clustering starts with every data point in its own cluster and then integrates the most comparable clusters periodically until a stopping criterion is met. The approach is then used to create a cluster hierarchy that finally leads to one cluster containing all of the data points [34]. This method creates a bottom-up hierarchy. The hierarchical structure is built using a "merge & agglomerate" technique. Several steps are involved in the agglomerative technique. First, it begins by treating every point of data as a different cluster. Then it creates a similarity or distance matrix between all clus-

ter combinations. Then two adjacent clusters are combined using the selected similarity and distance metric. The similarity and distance matrix is recalculated to account for the combined cluster. Steps 2, 3, and 4 are continued until all data points are assigned to a particular cluster or a preset stopping threshold is satisfied. A dendrogram may be used to illustrate the consequent hierarchical structure. Because it eliminates the requirement for recursion division, agglomerative clustering is more productive than divisional clustering [35]. However, it could suffer from the "chaining effect", in which early integrating decisions cannot be reversed, thus resulting in inferior clustering outcomes. There are a few algorithms that employ hierarchical clustering in various ways: BIRCH, CURE, ROCK, and CHAMELEON.

4.1.3. Balanced Iterative Reducing and Clustering with Hierarchies Method

BIRCH is a clustering approach that tries to cluster huge datasets effectively by reducing memory utilization and computational difficulties. Zhang et al. [36] presented it in 1996 as a hierarchical clustering approach. BIRCH is intended to manage big datasets by building a memory tree-like data structure known as the cluster feature (CF) tree. It depends on the ideas of the CT tree; CF is a triplet (n, LS, SS), wherein n is the number of data items in the clusters, LS is the linear product of the item's attribute values, and SS is the total number of squares of the item attribute values. These are stored in a tree known as the CF tree. Because it is kept in a tree, they are unable to retain entire tuples or clusters in the primary memory, only their tuples [37]. The BIRCH technique has a number of features. As BIRCH can deal with huge datasets effectively without needing the complete dataset to be put into memory, it makes it suited for resource-constrained applications. BIRCH also offers quick online developments, enabling the structure of the trees to be continually altered as unique data points come. Furthermore, the hierarchical structure of the clustering findings allows for varying degrees of precision in the clustering outcomes, enabling users to examine clusters at multiple levels of depth [38].

4.1.4. Clustering Using Representatives Method

CURE is a hierarchical clustering approach aimed at overcoming the constraints of typical hierarchical clustering techniques when dealing with massive datasets. Guha et al. [39] presented CURE in 1998 as a hierarchical clustering technique. CURE attempts to solve the reliability and effectiveness concerns associated with huge datasets in order to transcend the constraints of existing hierarchical clustering techniques. It is a large data clustering approach that is more resistant to outliers and obtains clusters of all forms and shapes. It works well with two-dimensional datasets. It has an $O(n^2 logn)$ operational complexity [40]. Both BIRCH and CURE manage outliers effectively. BIRCH has lower temporal complexity and lower cluster integrity than the CURE method.

CURE has various benefits when it comes to clustering huge datasets. It uses a sample-based strategy to manage datasets containing billions of data points, lowering processing needs. CURE's hierarchical structures allow for the exploration of clusters at various degrees of granularity. Furthermore, because it is not predicated on any certain cluster structure or dispersion, CURE may handle clusters of different sizes and forms [41].

4.1.5. Robust Clustering with Links Method

Robust clustering with links (ROCK) is a hierarchical clustering technique developed in 1999 by Rastogi and Shim [42]. ROCK is a density-based technique to find clusters in datasets that could have noise and outliers. It uses an agglomerative hierarchical clustering technique to cluster information based on categories. It depends on the number of connections that exist between two items; linkages represent the number of additional data that are adequately comparable to the two. This method does not use a distance metric. ROCK has a number of benefits like the capacity to manage datasets of varied densities and its resistance to noise and outliers. ROCK may locate clusters irrespective of a setting of asymmetrical clusters and noisy data by using the density and connection strength

measurements. ROCK, on the other hand, has several limits. The similarity function has a considerable influence on the clustering outcomes, and finding a good measurement of similarity for individual datasets may be difficult. The efficiency of the method is also determined by the variables used, like the neighborhood size and the minimal similarity criterion [43].

4.1.6. Clustering Heterogeneous and Attributed Multi-Modal Environments Using Localized Similarity Indicators Method

CHAMELEON is a hierarchical clustering approach for datasets with heterogeneity characteristics and numerous modalities. Karypis, Han, and Kumar presented CHAMELEON in 1999 as a hierarchical clustering technique [29]. It is built to manage datasets with varied densities, irregular forms, and changing sizes. It makes use of various similarity measurements and an adaptive merging method to enhance clustering performance. As it is a hierarchical clustering algorithm as well, two clusters are combined only if their interconnection and similarity (proximity) are significant in comparison to their internal interconnectedness and proximity of items inside the clusters.

The efficiency of the algorithm can be impacted by the similarity metrics used, and also by the clustering techniques used for the initial attribute partitioning. CHAMELEON, which is a computational strain of approach, grows with the extent of the dataset and the number of variables or modalities, which makes it better suited to small to moderate-sized data [44].

4.2. Density-Based Clustering

Density-based clustering is a type of algorithm for clustering that organizes data points in the space of features depending on their density. This technique was introduced by Kriegel et al. in 2011 [45]. Density-based clustering discovers clusters as territories with significant point density rather than as independent areas with specific features. Density-based clustering is a clustering method that organizes data points in the dataspace based on their density. It seeks to detect clusters of any shape or dimension by recognizing significant data density locations. The primary notion underlying density-based clustering is that clusters can be described as high-density areas bounded by lower-density regions. It makes no predictions on the number of clusters in the dataset or on preset cluster forms [46]. The following are some popular forms of density-based clustering algorithms.

4.2.1. Density Based Spatial Clustering of Applications with Noise

Rehman et al. introduced the density-based spatial clustering of applications with noise (DBSCAN) where clusters are defined as regions of high density that split from areas of low density. It is noisy and outlier-tolerant and can recognize clusters of any sort [47]. Because of its capacity to identify clusters of different forms and manage datasets of different densities, it has become one of the most prominent and commonly used methods for clustering. DBSCAN defines core, boundary, and noise points depending on the density of the dataset in the features area [48].

Another variant that offers a hierarchical approach to density-based clustering is hierarchical DBSCAN (HDBSCAN) [49]. It builds cluster hierarchies by organizing the data points into a tree-like framework that represents clusters of varying densities. The hierarchical model makes it easier to identify clusters at different granularity levels, resulting in a more thorough comprehension of the data's cluster pattern.

Categorical DBSCAN (CDBSCAN) expands DBSCAN to accept categorical characteristics in order to deal with categorical datasets [50]. By specifying distance and density criteria applicable to category similarity measurements, CDBSCAN extends the density-based clustering technique. This enables the identification of clusters in data with both numerical and categorical features.

Other DBSCAN versions and enhancements handle specific issues and circumstances. For example, k-DBSCAN extends DBSCAN by including how to choose features and

deal with large data and alleviate the affliction of dimensionality. The local outlier factor (LOF) [51] extension detects outliers in data by using local density. These modifications and enhancements illustrate DBSCAN's versatility and ability to adapt to various clustering circumstances. These improvements improve the algorithm's effectiveness and application in a variety of disciplines, like spatial analysis of data, identifying anomalies, and recognition of patterns, by combining new methodologies and adjustments.

4.2.2. Density-Based Clustering

H. Rehioui and A. Idrissi introduced the density clustering (DENCLUE) algorithm [52]. It is a density-based clustering approach that uses the concept of attractants to locate clusters. The data points are modeled as prospective attractors, and clusters are discovered in the vicinity of these attractors. The approach begins by utilizing a kernel function to estimate the density of every point of data depending on its local neighborhood. Density estimation considers the proximity and effect of neighboring locations, enabling DENCLUE to detect intricate patterns and neighborhood features

DENCLUE is an iterative technique to locate clusters by searching for density attractors and expanding clusters from them. It locates attractors using a gradient ascent technique that follows the sharpest ascending path in the predicted density matrix. Data points are allocated to clusters on the basis of their closeness and density interaction with the attractors as they are identified. The clustering procedure is repeated until no further attractors and clusters have been found [53].

X.-G. Yu and Y. Jian proposed a novel clustering technique by merging the concepts of K-nearest neighbors (KNN) and DENCLUE [54]. To find the initial cluster centers and allocate data points to their nearest neighbors, the method employs the KNN technique. The density-based clustering approach of DENCLUE is then used to improve the clustering outcomes using the local density attractors. The suggested technique seeks to increase the accuracy and resilience of clustering by taking into account both the local neighborhood links recorded by KNN and the density-based clustering properties of DENCLUE. The combination of this method has an opportunity to produce more accurate and complete clustering results, especially in datasets having complicated structures and changing densities.

4.2.3. Ordering Points to Identify Clustering Structure

Mihael Ankerst, Hans-Peter Kriegel, and Jörg Sander presented ordering points to identify clustering structure (OPTICS) in 1999. It is a modified version of DBSCAN; however, the input variable requirements are less stringent [55]. It generates a database placing orders, saving the core distance and a reasonable reachability distance for every item. A clustering framework is constructed that defines a wide range of potential values and clusters the data autonomously and dynamically. OPTICS determines an enhanced cluster ordering using data on a wide range of characteristics, similar to density-based clustering. This method has received changes and adjustments over time in order to increase its performance and meet certain clustering problems.

FastOPTICS, an improved version of the method that enhances the accuracy for huge datasets, is one improvement of OPTICS [56]. FastOPTICS makes use of a number of optimization approaches to minimize the computing complexities involved in evaluating accessibility distances and creating the OPTICS graphic. It enables quicker handling and evaluation of datasets containing countless data points. Another innovation is the combination of OPTICS with visualization approaches. Users may more successfully study and analyze clustering findings by integrating OPTICS with dynamic visualizations. Scatter graphs, heatmaps, and dendrograms may be employed to show the hierarchical structure and density distributions of the clusters, allowing for a more instinctive comprehension of the data.

4.3. Grid-Based Clustering

Grid-based clustering is a clustering approach that utilizes the data field as divided into grid cells, and data points are allocated to them. Hinneburg and Keim first used it in 1998. It can provide efficient and scalable clustering methods, which are extremely useful for large datasets [57]. Grid-based clustering works by dividing the dataspace into a standard grid of units. According to the kind of data being clustered, every cell indicates a geographical region or a collection of values associated with attributes. The grid units are then allocated data points depending on their precise position or value of an attribute [58]. The following are some popular grid-based clustering approaches.

4.3.1. STING

Statistical information grid (STING) is an acronym that refers to the statistical information grid. Bureva et al. introduced STING, a grid-based clustering technique, in 2017 [59]. To promote effective grouping and analysis of spatial data points, STING separates the data space into a hierarchical grid pattern. STING divides the data space into an irregular grid at first, and statistical data like mean, variance, and correlation are computed for every cell in the grid. The statistical analysis provides the distribution of data across each cell. Cells with similar statistical qualities are combined to produce bigger cells, leading to a grid structure. Clustering in STING is accomplished by assessing the uniformity of the statistical data contained within every cell. If a cell fails to match the stated homogeneity requirements, it is divided into several sub-cells. The method is repeated until all cells meet the homogeneity criterion or the appropriate degree of precision is obtained. It generates clusters using cell data and statistical approaches to determine cell similarities [60].

4.3.2. CLIQUE

Clustering in quest (CLIQUE) is an acronym that refers to clustering in the quest. Agrawal et al. presented it in 1998 as a grid-based clustering technique [61]. CLIQUE seeks to discover dense areas inside a fixed-size grid pattern referred to as cliques. It is a clustering algorithm that uses a grid-based framework to find the dense regions in the data field. Each component in the grid layout is represented by a maximal clique, while clusters are described as connected dense units. It divides the dataspace into fixed-size grid cells, and every cell in the grid is considered as a possible clique. Cliques are identified by analyzing the density of data points across every cell. If the number of points inside a cell reaches a predetermined density threshold, a clique appears. CLIQUE works from the ground up. It begins with each cell and combines nearby cells to form bigger cliques once the density conditions are met. The merging procedure is repeated until no further cliques may emerge [62].

4.3.3. GridDBSCAN

T. Boonchoo et al. proposed GridDBSCAN. It is a grid-based variant of the well-known DBSCAN technique. This technique begins by creating a grid structure that divides the data space [63]. A customized parameter, often the epsilon distance employed by DBSCAN, determines the capacity of the grid cells. Depending on their geographical supervision, the data items are subsequently allocated to the respective grid cells. GridDBSCAN then executes the DBSCAN core stages within every grid cell. It finds its center points that are placed with an adequate number of neighbors inside the epsilon distance. The technique grows clusters by linking core locations that have neighbors in general. Points that do not belong to any cluster are referred to as noise and outliers. It dynamically modifies the epsilon value for every cell in the grid depending on its density attributes. This enables dynamic density estimation, which takes into consideration local density fluctuations across various sections of the data space.

GridDBSCAN has enhanced scalability and lowered computing complexity when compared with the original DBSCAN technique. GridDBSCAN decreases the number of

distance computations required by separating the data space into grids, which makes it more effective for huge datasets [64].

4.4. Partitioning-Based Clustering

It is a type of clustering algorithm which divides datasets into distinct groups by optimizing a function with objectives. It continuously allocates data points to clusters and maintains the cluster centroid points until convergence [15]. This method allocates every data point to precisely one cluster, and the number of clusters is either given initially or decided by the technique dynamically. K-Means and K-Medoid are two popular partitioning-based clustering algorithms. K-Means and their related techniques and K-Medoid-related techniques are discussed here.

4.4.1. K-Means

K-Means clustering is an unsupervised learning technique. Clustering is used to group data items based on how comparable they are. The letter "K" is based on how comparable they are. The letter "K" in the K cluster data points are clustered using clustering [15]. Each data point's distance from the two center points has to be computed. The distance between the center points of every data point is determined, and the data item is then assigned to the center point with the shortest distance. This approach is used for each dataset to allocate it to a center point. Once the data points have been assigned to centroids, the next phase is to calculate the exact center point for each of the two clusters of data. To update the centroid of every cluster, the average of all the data items assigned to it is taken. It is repeated and each phase is updated iteratively until the centroids no longer change substantially or until the desired number of iterations is reached. The approach converges when the centroids stabilize and their locations cease shifting substantially. As the final outcome of the K-means technique, a set of K clusters with one data point each is produced. This clustering has been extensively used in a wide range of applications including market segmentation, image segmentation, and identifying anomalies. However it has several downsides such as vulnerability to the random assignment of the first centroids or the need for specialized knowledge to determine the optimal value of K. Despite these shortcomings, K-Means clustering remains a popular and successful approach for data analysis and pattern recognition.

K. A. A. Nazeer developed an improved version of the K-Means approach that involves sorting and partitioning of the dataset into "k" sets, resulting in better beginning centroids and hence boosting the algorithm's performance [65]. When compared to the traditional K-Means method, this strategy converges faster. The value of k (the needed number of clusters) must still be given as input, which may be tricky in some cases. This is one of the technique's major drawbacks.

L. Xumin developed an improved K-Means approach for dealing with the problem of calculating the Euclidean distance between each data point and all cluster centers in each iteration, which increases the running time [66]. Each iteration of this procedure keeps certain information in a structure of data that may subsequently be used in the next cycle. This dramatically decreases processing time, especially for large datasets with a large number of clusters. When evaluated against a variety of benchmark datasets, the proposed approach surpassed the standard K-Means algorithm in regard to accuracy and speed.

The convexity constraints variational K-Means (CV K-Means) approach was created by S. Ren and A. Fan to address the issue of unnecessary features caused by standard K-Means clustering's use of the Euclidean distance as a similarity metric [67]. A weight vector based on the coefficient of variation is provided to reduce the impact of insignificant attributes. The main disadvantage of this strategy is that it still requires input, namely the needed number of clusters (k).

Z. Zhang proposed an improved K-Means clustering algorithm that optimizes the initial centroids based on data dimension density [68]. The method ensures that the initial centroids have the most cluster-to-cluster variance. This technique is implemented on the

Hadoop platform using the Map-Reduce programming model. This strategy improved the stability of K-Means clustering.

4.4.2. Fuzzy C Mean

The fuzzy c-means (FCM) clustering algorithm is an unsupervised approach to data clustering. It is a K-Means variant that decreases the sum of the squared distances between cluster centroids and data points in each cluster. FCM assigns a fuzzy membership degree to each data point, representing how much it belongs to each cluster. Data points can belong to many clusters in part, allowing for more sophisticated categorization. The extensive use of FCM in data mining and machine learning benefits, in particular, image segmentation, pattern recognition, and informatics [69].

The FCM approach begins by randomly initializing the degree of membership values, and the number of clusters for every point of data. The centroid of each cluster is then determined using the membership degrees as weights. After that, the membership degrees of each data point are adjusted based on its distance from the centroids. It is iterated until convergence is reached or until a user-specified number of times (the iteration could become stuck at specific local maxima or minima). FCM has several applications including data mining, pattern recognition, and image segmentation [70]. It is an effective clustering method for datasets with ill-defined cluster boundaries.

T. Velmurugan [71] performed a comparison of K-Means and FCM clustering methods based on the number of samples and groupings. The findings show that K-Means outperforms FCM in general, as FCM requires more time to complete fuzzy measure computations, increasing its temporal complexity and influencing its outcomes. Although FCM gives outcomes similar to K-Means, its time complexity remains very high.

Banerjee, S. conducted a comparative analysis of multiple KM algorithm versions, such as Bisecting K-Means, FCM, and genetic K-Means [72]. Genetic K-Means exceeds the other clustering algorithms in terms of both internal and external indices and delivers the best performance, according to the data.

S. Ramathilagam developed an excellent fuzzy segmentation algorithm for breast magnetic resonance imaging (MRI) data using kernel-induced FCM, an objective function of FCM [73]. This technique's foundation is the hyper tangent function, which is based on the kernel function and Lagrangian multipliers. When compared to current fuzzy segmentation techniques on the same dataset, the suggested technique performed better in terms of precision, specificity, as well as accuracy. The approach may be used to precisely and effectively segment breast MRI datasets, which is critical for breast cancer diagnosis and therapy.

Huynh Van Lung and Jong-Myon Kim developed the generalized spatial Fuzzy C-Means clustering (GSFCM) technique, which is used for brain MRI segmentation [74]. GSFCM employs both pixel characteristics and spatial local information, with the weights of each neighbor determined by its distance properties. The approach tries to reduce the over-segmentation problem associated with conventional FCM through the use of spatial information, resulting in improved segmentation outcomes. According to the results of the experiments, GSFCM outperforms regular FCM when it comes to segmentation reliability and precision.

Gerald Schaefer and Abdul H. Sadka presented an FCM calculation using mean shift for skin lesion removal. They proposed an FCM target function that adds a mean area factor into the traditional FCM target function according to mean shift [75]. According to testing data, their system is capable of effectively extracting the borders of skin lesions.

4.4.3. K-Means++

K-Means clustering is a popular clustering algorithm that splits data into K groups based on their similarities. K-Means has the disadvantage of being sensitive to how the center points are initialized, which may result in poor clustering [76]. To address this

issue and improve clustering quality, K-Means++ adopts a more sophisticated centroid initialization.

The K-Means++ approach ensures that centroids are initialized at distant places, reducing the chance of empty clusters or numerous clusters linked to a single centroid. Because of this initialization stage, the centroids are equally distributed over the data space, reducing the possibility that the algorithm may become caught in a specific minimum [77]. K-Means++ and the regular K-Means method are essentially identical, with the exception of the initial step. With K-Means++, we may improve our findings and avoid poor clustering. As a result, K-Means++ is a popular clustering algorithm in a range of fields, like artificial intelligence, data mining, and image segmentation.

The steps of the K-Means++ clustering method are as follows [78]. To begin, select one of the data points at random to act as the initial center point. For each subsequent centroid, calculate the distance between every point of data and the earlier picked closest centroid (using the Euclidean distance). Create a weighted distribution of probabilities with each data point's weight proportionate to its square distance from the nearest center point. Select the next centroid by choosing a sample from the weighted distribution of probabilities created in the previous step. Repeat steps 2–4 until all k center spots have been picked. Determine the distances between each data point's centroid, and subsequently locate the point in the cluster with the nearest centroid. Find the new centroid by calculating the average of the data points assigned to every cluster. Steps 6 and 7 should be repeated until the center points stabilize and the data point cluster assignments remain consistent. The procedure delivers the k clusters' final center points in addition to the final data point distributions to clusters. It must be mentioned that the K-Means++ approach has superior initialization than the initially developed K-Means algorithm. Using a weighted distribution of probabilities, the approach can select more accurate starting center locations, perhaps producing better clustering results.

Z. Min and D. Kai-fei presented a way of dealing with the K-Means++ technique's restrictions. The method selects the cluster center with a small amount of variation as the first beginning point to lessen the influence of limited points and maximize the reliability and precision of the clustering findings. However, the proposed strategy has two major flaws. First, due to the intricacies of the method used, it may take longer [79]. Secondly, if there is a large amount of data, it may cause computational issues. Several concerns must be addressed before the approach may be used in real-world applications.

4.4.4. MinMax K-Means

The MinMax K-means technique developed by Georgios Tzortzis and Aristidis Likas aims to increase the robustness of the K-means method by minimizing the greatest distance between data points and the center points assigned to them [80]. Utilizing this method, K centroids are selected at random, data points are clustered by assigning them to the closest center point, the greatest distance between each data point and its chosen center point is calculated, and center points are calculated again by placing them in the center of their respective clusters. MinMax K-Means has been evaluated against kernel K-Means, fuzzy K-Means, and standard K-Means algorithms.

The MinMax K-Means approach begins by randomly selecting K centroids from the available dataset. In the subsequent stage, the Euclidean distance is employed for allocating every data point to the nearest centroid. In the next step, the greatest distance between each data point and its allocated centroid is calculated. The centroids are determined again in the subsequent stage by shifting them to the middle of their corresponding clusters. Steps two through four are performed until convergence is obtained or the maximum number of iterations is reached. The final result is obtained by organizing the data points with respect to their nearest center points [80]. The MinMax K-Means approach is designed to minimize the greatest distance between data points and the allocated center points, thus being more resistant to complex datasets with intersecting clusters than the classic K-Means

algorithm. This approach may be particularly useful in boosting the stability and precision of clustering techniques in situations where points of data are varied or noisy.

4.4.5. K*-Means

K-Means is a well-known unsupervised machine learning technique for clustering data. Although it is a simple and effective approach, there may be some performance concerns, such as susceptibility to the starting center points and a tendency to converge to inferior solutions. M. C. Hung proposed the K*-Means technique to address these shortcomings [81]. The K*-Means method improves on the standard K-Means algorithm by using a dynamic sampling technique for selecting initial centroids and an enhanced updating procedure for the centroid during each iteration. The technique operates by first utilizing a dynamic sampling strategy that takes into account both distance and weight, beginning with a small number of data points to act as the first centroids and set the number of clusters to K. To link all of the data points to the nearest centroids, the Euclidean distance is employed. The centroids are modified by computing the average of all the data items assigned to each central point. Steps 2 and 3 are repeated until convergence is reached.

K*-Means uses a dynamic sampling strategy to choose initial center points which takes into account both the distance between data points and the number of data points. The approach selects an arbitrary data point as the initial center point and then selects further center points based on their distance from the preceding center points and their neighborhood weight. The first center points are well-distributed and precise representations. The number of data points assigned to each center point has been taken into account by an altered modify rule employed by K*-Means to modify the centroids through each iteration.

4.4.6. K-Means with Classification Method

Arpit Bansal et al. propose a unique method that integrates the K-Means clustering algorithm and an approach for classification to improve the prediction of data accuracy [82]. The K-Means approach is used in the first stage to group similar data points into K clusters, and then each cluster is put through a classification approach such as a decision tree or Naive Bayes to anticipate the result. Utilizing real-world datasets, researchers compared the precision of the proposed approach to that of the standard K-Means algorithm. The results showed that the proposed technique performed better in terms of prediction. This concept offers an improved way of predicting results using data and may find applications in a variety of fields like business, medical care, and finance. Because it integrates the positive aspects of clustering and classification techniques, the proposed method provides more exact result prediction than standard clustering methods alone.

4.4.7. K-Medoid

K-Medoid has been investigated from several perspectives including initialization processes, distance metrics, optimization approaches, evaluation measurements, etc. Partition around medoids (PAM) was created in 1987 by Kaufman and Rousseuw [83]. PAM, as opposed to K-Means, depicts a cluster by its medoid, indicating the structure that occurs most centrally situated inside the cluster. Medoids can be noisier and more outlier-resistant than center points. It was created to overcome the flaws of the K-Means algorithm. It selects K medoids randomly or by a heuristic approach and data are assigned to a cluster using a distance metric. The data point inside every cluster is chosen with the smallest dissimilarity as the subsequent medoid, substituting the prior medoid. The delegated and update stages are repeated until a stopping requirement is reached. This might be a set number of repeats, medoid location convergence, or a decrease in total dissimilarity [28]. When the algorithm convergence occurs, the resultant clusters are made up of data points that have been allocated to their corresponding medoids. Throughout each cluster, the medoids reflect the most prominent or significant areas.

Tagaram attributes clustering as an unsupervised method of learning which enables us to split data into groups [84]. Velmurugan et al. described clustering as an unsupervised

learning process [85]. According to these studies, the K-Medoids approach is more resilient than K-Means clustering when it comes to noise and outliers, although it is only suitable for the smallest datasets.

4.4.8. Combining K-Medoids Clustering and Naïve Bayes Classification

After the data items have been classified into clusters utilizing K-Medoids clustering, Naive Bayes (NB) classification is performed to allocate labels to the clusters [86]. This integrated approach produces better results. An NB classifier is built using the retrieved features and the matching labeled classes. The NB method is based on the assumption of feature isolation and predicts the conditional probability of every attribute given the class labeling. To identify fresh, previously unknown data points, a trained NB classifier is used for classification. The most possible class label is determined using the retrieved characteristics from each data point and the learned distributions of probability [87]. Relevant metrics are used for assessing the combined clustering and classification outcomes like performance, precision measure, recall, and F1 score. The performance of the approach is analyzed along with any required modifications or enhancements.

4.4.9. Partitioning around Medoids and Its Variants

The partitioning around medoids [88] is composed of two algorithms: BUILD to select a starting point for clustering and SWAP to enhance the clustering at a local optimum (locating the optimal global value of the K-Medoids issue is regrettably NP-hard, as demonstrated by Kariv and Hakimi [89]). The techniques need a dissimilarity metric (which can be determined using Kaufman or Rousseeuw's regular DAISY, which needs $O(n^2)$ memory and usually $O(n^{2d})$ time to determine, but possibly much more for prohibitive distances like earth mover's distance, also referred to as Wasserstein metric). In numerous situations, calculating the distance matrix has become a bottleneck [90].

The BUILD technique, introduced by Z. Li, G. Wang, and G. He, identifies an initial collection of medoids and gives a solid initial point for the remaining iterative phases of PAM [91]. Initially, the total variance of the data item to all presently selected medoids is calculated. The medoid linked with the lowest overall dissimilarity value is considered. The data point with the lowest overall dissimilarity is chosen as the subsequent medoid after considering the variance for all data points. The BUILD method iteratively examines every data point's dissimilarity to the present set of medoids and chooses the data point with the lowest overall variance as the subsequent medoid. This procedure guarantees that the first medoids are selected based on their capacity to accurately reflect the data. After selecting the first medoids using the BUILD method, PAM goes through the allocation and update processes to repeatedly modify the medoids, thus improving the clustering result.

The SWAP optimization approach is used in PAM to enhance the clustering approach by iteratively exchanging a medoid against a non-medoid, indicating and analyzing the resultant dissimilarity. The SWAP method, developed by H. Song and J.-G. Lee, tries to identify superior medoid allocations that minimize overall dissimilarity across clusters [92]. Begin using the initial collection of medoids acquired via the BUILD initialization process. Utilizing the dissimilarity metric, often the distance between the data point and the medoid, every data point is allocated to the closest medoid. Iteration over all medoids is carried out to investigate possible swaps and assess their influence on the clustering approach. The swap operation and assessment procedures are continued until no additional swaps reduce overall dissimilarity. The approach has reached convergence at this stage, so the final collection of medoids reflects a better clustering result [93].

4.4.10. Clustering Large Applications

Clustering large applications (CLARA) is a clustering algorithm designed specifically for handling large datasets. It integrates the K-Medoids clustering approach with a sampling-based mechanism to successfully cluster large datasets. CLARA is a clustering technique optimized for huge datasets. It is a K-Medoids clustering method modification

that overcomes the scalability concerns associated with classic K-Medoids techniques. CLARA splits the dataset into numerous independent samples and uses K-Medoids clustering to produce representative medoids for every single sample. S. Renjith and A. Sreekumar provide the CLARA method, which first generates a large number of samples at random from the initial set of data. The quantity and amount of samples are governed by the computing resources provided [94]. To choose a collection of medoids, the K-Medoids technique is run on every single sample. This is accomplished by modifying the medoids iteratively in order to minimize the overall dissimilarity and distance between the medoids and the data points inside each sample.

The medoids collected from the samples are examined for quality. This is accomplished by determining the mean variance or distance between each medoid and the data points in the total dataset. According to their assessment ratings, the most effective medoids from the samples are chosen. These medoids reflect the clusters' final collection of relevant data points. On the basis of a distance measure of your choice, each data point in the dataset is allocated to the closest medoid. Every point of data is assigned to a cluster that includes its closest medoid. Relevant metrics are used to assess the clustering results, like cluster purity, silhouette coefficient, or various domain-specific indicators. The clusters and the clustering method's patterns or discoveries are analyzed. CLARA overcomes the limits of K-Medoids approaches in big dataset management by employing random collection and medoid selections from numerous samples [95]. CLARA has the ability to deliver scalable and effective clustering techniques for applications with enormous volumes of data.

4.4.11. Clustering Large Applications Based on Randomized Search

R. T. Ng and Jiawei Han proposed the clustering large applications based on the randomized search (CLARANS) technique. CLARANS explores the solution domain through a randomized search using various combinations of medoids [95]. CLARANS starts by picking K medoids arbitrarily from the dataset. For every iteration of CLARANS, a neighbor medoid is chosen arbitrarily for every current medoid. The number of neighbors to take into account is a customized parameter that governs the search's precision. CLARANS conducts a local search by exchanging the present medoid for the neighbor medoid and assessing the goal function (such as total dissimilarity) for the resulting medoid set. The goal is to enhance or optimize the clustering approach by lowering total dissimilarity and another criterion. If the desired function value changes or meets a preset threshold, CLARANS approves the updated medoid set. If a substitute medoid set is approved, it becomes the present medoids' updated set; alternatively, the swap is denied, while the existing medoids stay unaltered. CLARANS iterates through neighbor research, local search, and accept or reject processes a predetermined number of times. When the required number of iterations is achieved, the algorithm terminates. As the ultimate clustering approach, CLARANS delivers the most effective medoid set discovered over the rounds. The number of iterations and the greatest number of neighbors are searched to determine the level of accuracy of the result.

Instead of examining all potential swaps, CLARANS employs a randomized search. It selects an arbitrary combination of a non-medoid item and a medoid item, determines when this enhances the present loss, and subsequently greedily conducts the swap. Relating the FastPAM1 notion to CLARANS' arbitrary exploration technique, just the non-medoid item is chosen at random, and all medoids are examined for switching at a cost comparable to searching at one medoid [96]. This implies that we may either investigate K times the number of vertices of the graph or minimize the number of samples that are required by a factor of K. The second option is selected to achieve results comparable to the distinctive CLARANS in terms of the number of edges examined; however, because the edges selected require exactly the same non-medoids, a small decrease is anticipated in quality, which can be easily compensated for by increasing the non-medoids subsampling rate. The consumer may easily alter the balance between calculation time and exploration by changing the sub-sampling rate option.

CLARINS is a clustering technique that is optimized for handling huge datasets. To discover the best medoids and clustering solution, it employs a randomized search technique. Wei et al. discovered that genetic approaches worked only for short datasets, small K, and well-separated symmetric clusters and that CLARANS was typically superior [97]. The CLARANS method views the searching space as a high-dimensional hypergraph, with each edge representing the swapping of a medoid or a non-medoid. The technique may be used to effectively investigate the K edges relating to all medoids at the same time; this enables us to investigate a bigger portion of the search area in the same amount of time, although we estimate the savings to be rather minimal in comparison to the gains obtained in PAM.

Table 3 provides a comprehensive overview of different clustering techniques and their applications

Table 3. Different clustering techniques and their applications.

Technique	Advantages	Limitations	Applications
Hierarchical clustering [29,32]	• It is not necessary to determine the no. of clusters. • Retrieves the hierarchical relationships that exist between clusters. • It allows for the rapid and easy identification of noisy dataset and outliers.	• Large datasets are substantially more costly. • It responds to original conditions and may be challenging to manage data with several properties. • It is difficult to determine the optimal no. of clusters for analysis.	• Consumer segmentation and market analysis. • Pattern and image recognition. • Text mining and document clustering.
Density-based Clustering [46]	• It is capable of detecting any type of cluster. • Robustness to noise and outliers. • It is capable of managing a wide variety of cluster densities. • Suitable for datasets with a biased or uneven distribution	• Sensitive to parameter selection, such as density and distance criterion. • Difficulty in effectively handling high-dimensional datasets. • Scalability issues arise with large datasets because of density calculations. • Identifying clusters with a little varying density is difficult. • When handling varied densities of data, performance degrades.	• Recognition of items and image segmentation. • Analysis of credit card transactions and fraud detection. • Detecting anomalies and identifying outliers in many disciplines.
Grid-based clustering [58]	• Scalable and efficient for large datasets. • Simple and easy to implement. • Capable of dealing with data of varying density. • Creates a grid structure to help you detect spatial patterns and visualize groupings. • Lower complexity in contrast to distance-based approaches.	• Changes in grid size and resolution can have an impact on clustering results. • Difficulty managing clusters spanning many grid cells. • Grid orientation and data dispersion sensitivities. • Capability to manage groups of irregular forms is limited. • Incapability to capture complex cluster interactions with enough flexibility.	• In computer vision and image processing, cluster analysis is used. • Fraud and outlier detection in massive datasets. • Detecting and analyzing network intrusions. • Clustering is used in exploring data, analysis, and mining.

Table 3. *Cont.*

Technique	Advantages	Limitations	Applications
Partition-based clustering [15]	• Partitioning methods are frequently computationally effective and scalable, which makes them appropriate for huge datasets. • They can work with a variety of data formats, including numerical and categorical properties. • Partition clustering allows for simple interpretations because every point of data corresponds to just one cluster. • Partition clustering approaches, such as k-means, have received substantial research and widespread use, with widely recognized algorithms along with evaluation criteria. • Partition clustering techniques often converge rapidly, making them computationally efficient.	• It is sensitive to starting points, resulting in varied results with different starting positions. • The user must define the number of clusters ahead of time, which might be difficult if the ideal number is unknown. • Partition clustering techniques imply spherical and similar-sized clusters, restricting their usefulness for complicated and non-linear cluster forms. • These methods are susceptible to outliers that can have a major impact on the creation of clusters.	• Customer Segmentation in Marketing • Image Compression and Segmentation in Computer Vision • Market Basket Analysis in Retail • Document Clustering in Text Mining

4.5. Comparison of Clustering Algorithms

Note that in Table 4, there is a comparison of different algorithms with respect to the following:

i. **Complexity**: Indicates the technique's computational complexity, where n is the no. of data points, k is the no. of clusters, I is the no. of iterations, and d is the data's dimension.
ii. **Cluster Shape**: Specifies the types of cluster forms that may be handled by the clustering technique.
iii. **Dataset Size**: Specifies whether the technique is appropriate for large-scale, small-scale, or both datasets.
iv. **Accuracy/Performance (Final Results)**: Specifies the technique's overall accuracy/performance in terms of the final clustering outcomes.

4.6. Hyperparameter Tuning in Clustering Algorithms

The importance of hyper-parameters in the context of Artificial Intelligence (AI) algorithms cannot be overemphasized. The course of algorithms can be directed by these seemingly harmless parameters to either outstanding performance or abject failure. In keeping with this crucial idea, this study launches a focused investigation of hyper-parameter optimization, a crucial subject that supports the effectiveness of AI systems. In addition, the mathematical formalization of hyperparameter optimization (HPO) is fundamentally a black-box optimization, frequently in a higher-dimensional space, so it is preferable to outsource this to suitable algorithms and machines to boost productivity and guarantee reproducibility [98].

Table 4. Comparison of different clustering algorithms with respect to accuracy and their limitations.

Algorithm	Complexity	Cluster	Accuracy (Final Result)	Dataset Size	Limitations
K-mean [15]	$O(n*k*I*d)$	Spherical	Moderate	Large or Small	Sensitive to initial centroid selection, assumes equal-sized and density clusters
K-Mean++ [77]	$O(n*k*I*d)$	Spherical	Moderate	Large or Small	Sensitive to initial centroid selection, improves initialization over K-means
K-Mean* [81]	$O(n*k*I*d)$	Spherical	High	Large or Small	Robust to initial centroid selection, enhances K-means
Combining K-medoid and NB [87]	$O(n*k*I*d)$	Arbitrary	High	Small	Computationally expensive for large datasets
K-Medoid [28]	$O(k*I*n^2)$	Arbitrary, varying	Moderate	Small	Computationally exhaustive for large datasets
CLARA [94]	$O(s*k*I*d)$	Spherical, Fuzzy	Moderate	Large	Computationally expensive, limited to small datasets
CLARANS [96]	$O(n*k*I*d)$	Spherical, Fuzzy	Moderate	Large or Small	Computationally expensive, randomization improves results
PAM [90]	$O(k*I*n^2)$	Arbitrary, Varying	High	Small	Computationally expensive for large datasets
Minmax K-Mean [80]	$O(n*k*I*d)$	Spherical	High	Large or Small	Sensitive to outliers, non-linear scaling
Agglomerative Hierarchical [34]	$O(n^3*d)$	Arbitrary	Moderate	Small	Computationally expensive for large datasets
Divisive Hierarchical [32]	$O(n^3*d)$	Arbitrary	Moderate	Small	Computationally expensive for large datasets
BIRCH Hierarchical [37]	$O(n*I*d)$	Balanced	Moderate	Large	Sensitive to initial cluster centers
CURE Hierarchical [39]	$O(n*I*d)$	Arbitrary	Moderate	Large	Sensitivity to order of data points
CHAMELEON Hierarchical [29]	$O(n*I*d)$	Network	Moderate	Large	Sensitive to initial clusters, lacks robustness
DBSCAN [47]	$O(n^2*d)$	Arbitrary	High	Large or Small	Sensitive to density and distance parameters
DENCLUE [52]	$O(n^3*d)$	Fuzzy	High	Large or Small	Computationally expensive for large datasets
OPTICS [55]	$O(n^2*d)$	Arbitrary, Varying	High	Large or Small	Sensitive to density and distance parameters
STING grid [59]	$O(n*I*d)$	Arbitrary	Moderate	Large	Sensitive to grid size, assumes grid structure
CLIQUE [61]	$O(n^{(d+1)}*(m+logn))$	Network, Arbitrary	High	Small	Sensitive to density, parameter tuning required
Fuzzy C-Mean [70]	$O(n*I*c*d)$	Fuzzy	Moderate	Large	Sensitive to initial cluster centers

While the fundamental algorithms and architectures receive a lot of attention, hyper-parameters frequently go unnoticed in the process of fine-tuning these models for the best results. Therefore, a complete viewpoint is required, emphasizing the potential consequences of ignoring hyper-parameter tweaking. It is necessary to pre-specify an additional hyperparameter, which controls the size of the localization, in order to obtain improved clustering performance in particular applications [99]. There are many challenges in tuning hyperparameters for clustering algorithms, and these difficulties become more pronounced when we look at the particular methods discussed above.

- Algorithm Complexity: The search for optimal settings is difficult and time-consuming since many clustering algorithms, including K-means variants, hierarchical approaches, and density-based methods, contain sophisticated parameter interactions.
- Cluster Size and Shape: The dataset's cluster distribution, size, and shape can have a significant impact on how well the clustering algorithms function. With datasets that depart from these assumptions, algorithms like K-means, which assume spherical clusters and equal-sized densities, may have trouble.
- Sensitivity to Initializations: Several algorithms are sensitive to the initial placement of cluster centroids, including K-means and its variants (K-Means++ and K-Means*). Sub-optimal or even premature convergence to local optima might result from poor initiation.
- Computationally Expensive: For large datasets, some methods, such as density-based approaches (OPTICS, DBSCAN) and hierarchical approaches (Agglomerative Hierarchical, Divisive Hierarchical), may be computationally expensive. Such situations necessitate careful consideration of computational resources when tuning hyperparameters.
- Accuracy and computational cost trade-offs: The efficiency of clustering algorithms is sometimes sacrificed for accuracy. More precise algorithms could be computationally taxing, restricting their application to bigger datasets.

Recent studies have explored cutting-edge methods for hyper-parameter optimization, realizing how important it is to improve the efficiency of AI algorithms. The work of Calik et al., which uses regression surrogates based on deep learning to precisely characterize microwave transistors, is a noteworthy example [100]. This work not only demonstrates the effectiveness of cutting-edge methods but also emphasizes the necessity of precise hyper-parameter settings for accurate characterization outcomes. Karaman et al. take a particularly perceptive step when they promote the use of the artificial bee colony (ABC) algorithm for hyperparameter optimization. This project, aimed at real-time autonomous polyp identification of colorectal cancer (CRC), emphasizes the need for context-aware and specialized hyper-parameter settings. The idea that no algorithm, no matter how sophisticated, can thrive in the absence of painstakingly calibrated hyper-parameters is reinforced when this study is compared to the larger AI landscape [101]. A. Thielmann et al. examine the use of coherence, which measures the semantic consistency of a document's words, to improve the document clustering process of hyperparameter optimization. Traditionally, hyperparameter optimization entails fine-tuning algorithmic parameters to produce optimum results. The authors suggest a novel method for optimizing hyperparameters that incorporates coherence metrics [102], investigating novel strategies to handle hyperparameter adjustment in clustering algorithms in light of the aforementioned difficulties.

- Sensitivity Analysis: By methodically assessing how each unique hyperparameter affects clustering results, sensitivity analysis identifies the most important factors and successfully directs the tuning process.
- Metaheuristic algorithms: To efficiently search the hyperparameter space and identify optimal or nearly optimal configurations, strategies like genetic algorithms and particle swarm optimization can be used.
- Multi-objective Optimization: By simultaneously taking into account several goals, such as accuracy and computational efficiency, solutions can be found that strike a balance between these competing purposes.
- Transfer Learning: Accelerating the tuning process and enhancing the robustness of clustering algorithms can be accomplished by utilizing pre-tuned hyperparameters from comparable datasets or jobs.
- Automated Hyperparameter Tuning: Automated Machine Learning (AutoML) platforms can be used to speed up the process of hyperparameter tuning by using meta-learning and optimization algorithms to iteratively look for acceptable configurations.

It is clear that hyperparameters are important in deciding how well they work. Despite the complexity of these algorithms, poor hyperparameter selection might produce subpar outcomes. To address the difficulties posed by hyperparameter tuning, new strategies are

emerging. These include sensitivity analysis, metaheuristic algorithms, multi-objective optimization, transfer learning, and automated tuning. It is crucial to take the time and effort necessary to choose hyperparameters that are in line with the characteristics of the dataset and the needs of the algorithm if one is to derive meaningful insights from the data and obtain accurate clustering results.

4.7. Evaluation Measures in Clustering

Clustering evaluation measures assess the quality of clustering results by comparing the obtained clusters with known ground truth or by analyzing the internal structure of the clusters. Here are some commonly used evaluation measures in clustering.

4.7.1. Entropy

When ground truth labels are accessible, entropy-based methods like normalized mutual information and adjusted mutual information are extensively used strategies for assessing clustering performances. These metrics evaluate the exchange of data between anticipated clusters and genuine labeling of classes while accounting for the distribution of classes and cluster purity. They give an empirical assessment of the coherence between clustering and real labeling of classes, taking into consideration class and cluster distribution [103]. Figure 8 shows the impact of the number of clusters on entropy.

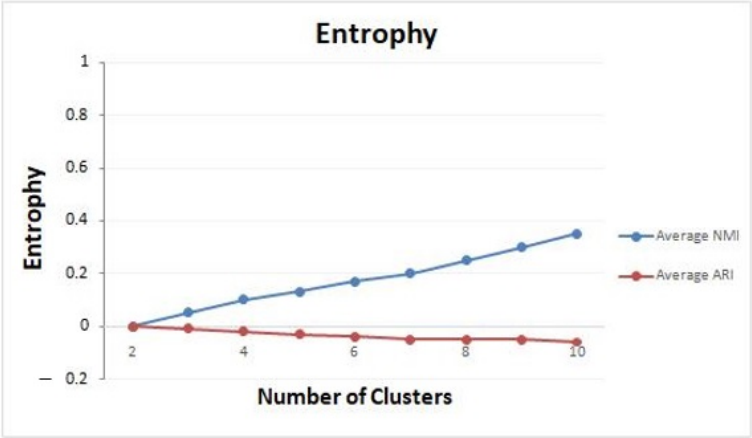

Figure 8. Entropy with respect to the number of clusters.

4.7.2. F Measure

The F measure is a popular clustering assessment metric that examines the precision and recall of clustering algorithms. To assess clustering efficiency, it delivers a single number that integrates these two criteria [104]. The F measure is used as an assessment metric in clustering, as follows.

Precision

Precision in the framework of clustering evaluation relates to the integrity of the clusters. It is the percentage of data points in a cluster that actually correspond to that group, without taking into account points from other clusters [105]. It evaluates how effectively a cluster depicts a certain group or notion.

Recall

The thoroughness of the clusters is referred to as recall in clustering evaluation. The percentage of data points from a certain class that are successfully allocated to the relevant cluster is measured by recall [105]. It evaluates the extent to which a cluster collects all of the data points in a particular class.

Harmonic Mean

Employing the harmonic mean, the F measure integrates accuracy and recall into one value.

$$HM = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (1)$$

The above equation is used to calculate the harmonic mean [106]. The F measure, by employing the harmonic mean, strikes a compromise between precision and recall, giving equal weight to both measurements.

Cluster-level F Measure

The F measure is determined at the cluster level in the clustering assessment. Every cluster is treated as a separate class, and actual class names are not necessary. The accuracy and recall for every cluster are calculated using the combination of data points allocated to that cluster and data points allocated to other clusters. Figure 9 shows how F measure varies with respect to the number of clusters.

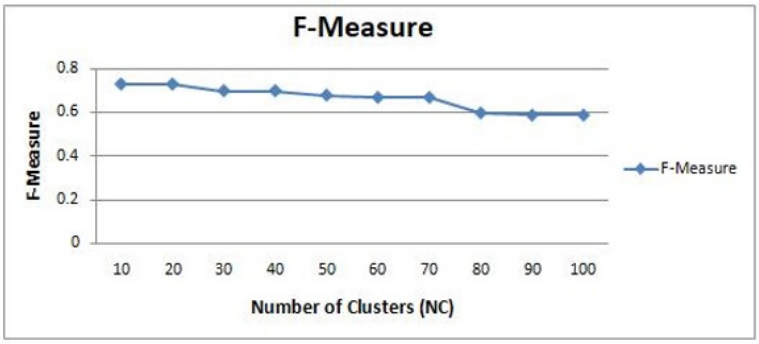

Figure 9. F measure vs. the number of clusters.

4.7.3. Rand Index

The Rand index is a popular clustering assessment metric that compares the similarity of two data divisions, like clustering results and actual class labels. It gives a quantifiable measure of the coherence between anticipated and true clusters, independent of the utilized clustering technique [107]. The Rand index is employed as an assessment metric in clustering as follows. Rand index against different numbers of clusters is shown in Figure 10.

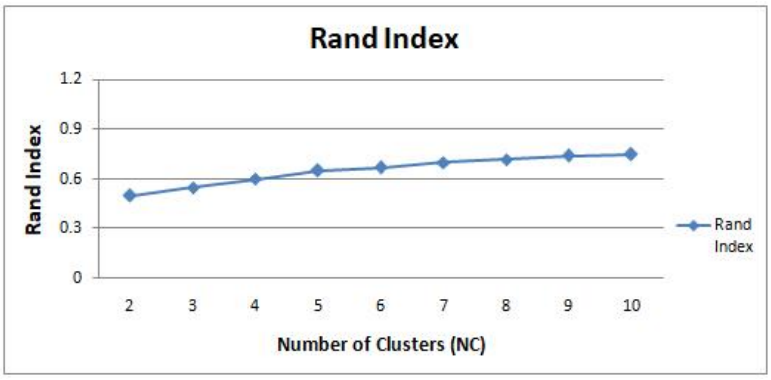

Figure 10. Number of clusters and Rand index.

4.7.4. Silhouette Coefficient

The Silhouette coefficient is a frequently used clustering assessment metric that evaluates the accuracy of clustering findings by measuring data point cohesiveness and separation inside and across clusters. It returns a single result for the complete clustering solution, showing how successfully the data items have been allocated to the appropriate

clusters [108]. The Silhouette coefficient is employed as an assessment metric in clustering as follows.

Cohesion

The degree to which data points inside a cluster are related is referred to as cohesion. It computes the mean distance between two data points inside the identical cluster. Lower cohesiveness means that the data items in a cluster are more densely packed and closer together [109].

Separation

The separation of the clusters relates to how different they are from one another. It calculates the mean distance between a data item and other data points in the same cluster. Higher separation suggests that the clusters are properly separated and distinctive. Silhouette coefficient index against different numbers of clusters is illustrated in Figure 11.

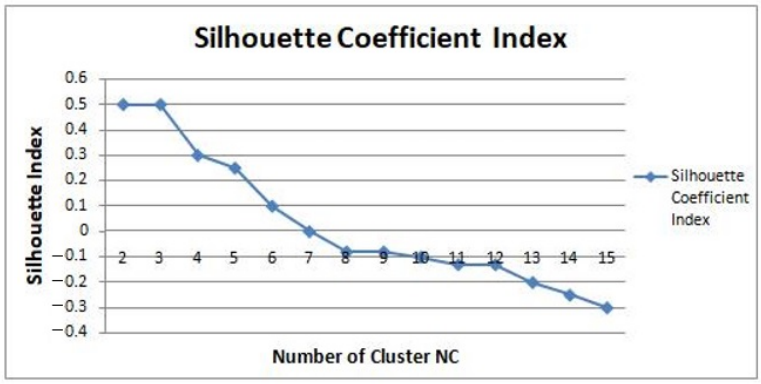

Figure 11. Silhouette coefficient index.

4.7.5. Dunn Index

The Dunn index is a clustering assessment metric that assesses cluster compression and separation based on inter-clustering and intra-clustering distances. It returns a single number indicating the level of accuracy of clustering findings. The Dunn index seeks to maximize the ratio of inter-clustering distance to intra-clustering separation [110]. The Dunn index is employed as an assessment metric in clustering as follows.

Inter-Cluster Distance

The variation and distance between various clusters are measured by inter-cluster distance. It indicates the distance or dispersion between clusters, demonstrating how distinct they are from one another. A greater inter-clustering distance indicates higher cluster separation [111].

Intra-Cluster Distance

The similarity and distance between data items inside the identical cluster are measured by intra-cluster distance. It denotes cluster compaction or cohesiveness, showing how closely data points are clustered inside each cluster. A lower intra-cluster distance indicates higher cluster cohesiveness [111].

Dunn Index Calculation

The Dunn index is determined by dividing the shortest inter-cluster distance by the shortest intra-cluster distance. The smallest distance across the two data points from distinct clusters is used to determine the inter-cluster distance. The greatest distance between the two data points inside the same cluster is determined as the intra-cluster distance. The Dunn index seeks to maximize this ratio, with a greater value indicating an improved clustering pattern. The impact of the number of clusters on the Dunn index is shown in Figure 12.

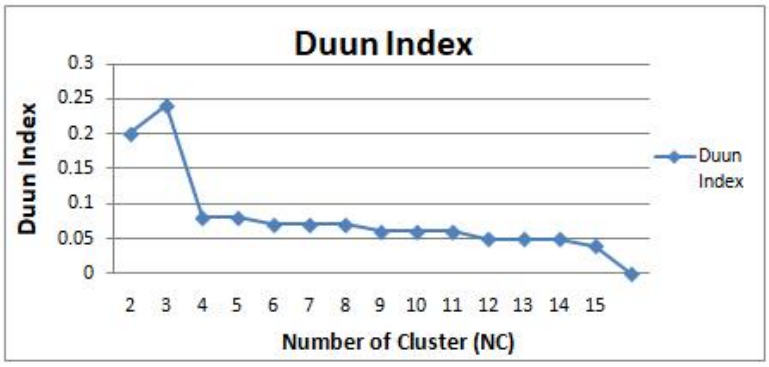

Figure 12. Impact of the number of clusters on Dunn index.

4.7.6. Davies–Bouldin Index

The Davies–Bouldin index is a clustering assessment metric that measures the accuracy of clustering findings based on intra-cluster and inter-cluster disparities. It returns a single result that evaluates clustering efficiency by taking into account both cluster compactness and cluster separation [112]. The more effective the clustering solution, the smaller the Davies–Bouldin index. The Davies–Bouldin index is employed as an assessment metric in clustering as follows.

Intra-Cluster Dispersion

Intra-cluster dispersion quantifies the distribution or scattering of data points inside each cluster. It measures cluster compaction and tightness, reflecting how tightly data points are packed inside each cluster. Lower intra-cluster dispersion means stronger cluster cohesiveness [113].

Inter-Cluster Separation

The variation or distance between various clusters is measured by inter-cluster dispersion. It measures the gap or distance of clusters to determine how distinct they are from one another. Greater inter-cluster dispersion means stronger cluster separation [113]. The Davies–Bouldin index against different numbers of clusters is given in Figure 13.

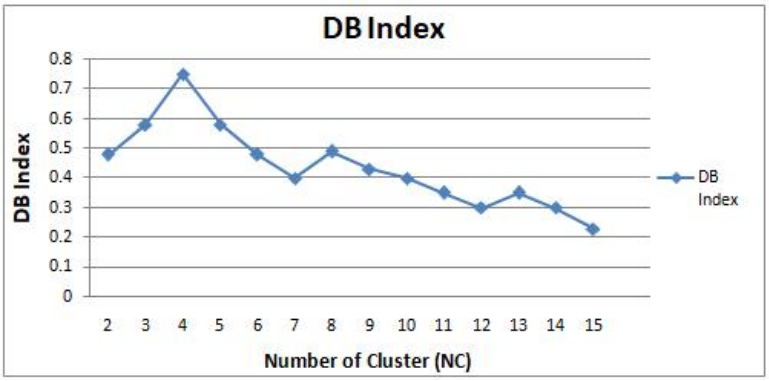

Figure 13. DB index vs. the number of clusters.

4.7.7. Calinski–Harabasz Index

This index calculates the ratio of variation between clusters to variation within clusters. It measures cluster cohesiveness and segregation, with greater values representing more accurate clustering. This index is very helpful when analyzing clustering techniques with predetermined cluster numbers [114]. The Calinski–Harabasz index is calculated by dividing the between-cluster variance by the within-cluster variance and multiplying the result by the ratio of $(N - K)$ to $(K - 1)$, in which N is the overall number of data points and K is the total number of clusters. This scaling element takes into consideration the number of clusters and the extent of the dataset when calculating the index. Figure 14 shows the Calinski–Harabasz index.

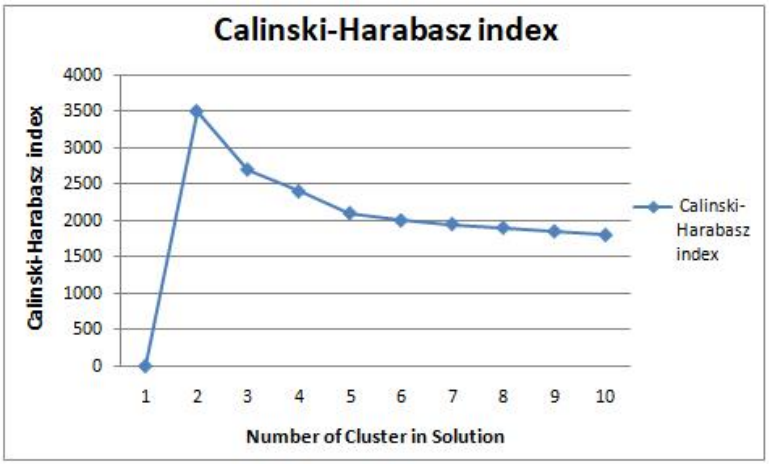

Figure 14. Calinski–Harabasz index.

4.7.8. Fowlkes–Mallows Index

It is a statistic used to compare the similarity of two clusters. It estimates the consensus between groups derived from two independent clustering findings, taking both the true positive and true negative allocations into consideration [115]. The formula of FMI is

$$FMI = \sqrt{(Precision * Recall)} \quad (2)$$

The FMI scales from 0 to 1, with 1 indicating total concurrence between clustering and 0 indicating no consensus beyond probability.

4.7.9. Hubert's Gamma Statistic

It is an assessment metric for determining the degree of coherence or connection between two distinct clusterings. It considers combinations of data points that appear in identical or distinct clusters in both clusters, offering a measure of coherence that goes beyond probability. It is computed using two elements: P and Q. P is the number of consistent pairings that have data points in an identical cluster in both clusterings or within distinct clusters in both clusterings. Q is the number of inconsistent pairings that have data points in the identical cluster within one clustering but in distinct clusters in another, or vice versa [116]. Table 5 shows the evaluation measures used in clustering along with their pros and cons.

4.8. Applications of Clustering

Clustering is an important process in data mining and artificial intelligence that involves grouping comparable objects or data items together based on their intrinsic features. Clustering has achieved tremendous progress in various fields of health, bioinformatics, social networking, etc. Figure 15 depicts the many fields where clustering may be applied.

4.8.1. Image Segmentation

Image segmentation is a common use for clustering algorithms such as K-Medoid and other clustering approaches. Picture segmentation is the process of dividing a picture into distinct parts or segments according to criteria like color similarity, appearance, intensity, or geographical closeness. Clustering techniques may be used to group together similar pixels as well as to distinguish various regions or objects in a picture [70]. Below is how clustering is used to segment images.

Table 5. Evaluation measures in clustering.

Metric	Calculation Method	Interpretation	Pros	Cons
Rand Index [107]	$\frac{(TP+TN)}{(TP+FP+FN+TN)}$	Consensus between two clustering's	Takes into account both positive and negative conditions.	Sensitive to the amount of clusters and combinations of data points.
Fowlkes–Mallows Index [115]	$\sqrt{(Precision \times Recall)}$	Similarities of two clusterings	Takes into account real positive and true negative decisions.	Does not consider cluster formation or general structure.
Silhouette Coefficient [108,109]	$\frac{(B-A)}{max(A,B)}$	Cluster density and segregation	Retrieves cluster coherence as well as dissociation.	Distance-based clustering is the only option.
Davies–Bouldin Index [112,113]	$\frac{(Ri+Rj)}{d(Ci,Cj)}$	The average extent of similarities between clusters.	Assesses cluster dispersion and compactness.	Relies on convex and isotropic clusters.
Calinski–Harabasz Index [114]	$\frac{\text{Between-Clusters Dispersion}}{\text{Within-Clusters Dispersion}} \times \frac{((N-K)}{(K-1)}$	Cluster segregation and compactness.	Modifies for the total number of clusters and the extent of the dataset.	Considers that clusters are spherical and of identical size.
Dunn Index [110,111]	$\frac{min(d(Ci,Cj))}{max(dia(Ck))}$	Cluster dispersion and density.	Measurement that is easy to understand.	Susceptible to noise and outliers.
F-measure [104]	$2 \times \frac{(Precision \times Recall)}{(Precision+Recall)}$	Recall as well as precision must be balanced.	Specifies a single metric that takes recall as well as precision into account.	It is determined by the threshold and similarity metric used.
Entropy [103]	$\sum(p \times logof(p))$	In clustering assignments, the degree of chaos or ambiguity.	Keeps cluster variety and heterogeneity.	Dataset probabilities must be estimated.
Hubert's Gamma Statistic [116,117]	$\frac{(P-Q)}{(P+Q)}$	There is a connection between two clusterings.	Allows for agreement in ways other than chance.	Sensitivity to the amount of clusters and repetitions of data points.

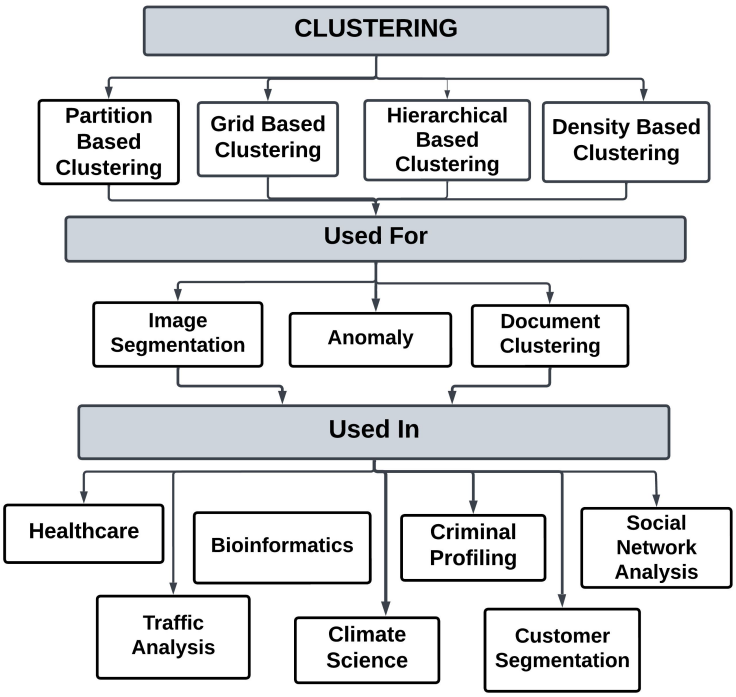

Figure 15. Applications of clustering in different domains.

Color-Based Segmentation

Color information from each pixel is utilized as a feature for clustering in color-based picture segmentation. Clusters are formed from pixels having similar color values. By considering pixel colors as feature vectors and grouping them into K groups, K-Means

and K-Medoid clustering may be used. Each cluster in the graphic indicates a section or area [118].

Feature Extraction

Color is not the only characteristic that can be taken from a picture; texture, intensity, and edge information may all be recovered. These properties, in conjunction with pixel color values, can be utilized to generate high-dimensional feature vectors. Clustering methods, such as K-Means, may then be applied to these feature vectors to group together similar pixels or areas [119].

Iterative Optimization

The K-Means and K-Medoid methods optimize the cluster centroids repeatedly in order to minimize the distance between each pixel and its assigned centroid. Within each cluster, this optimization procedure seeks the optimum representation of the color or feature distribution.

Post-Processing

Post-processing methods can be used to improve the segmentation results after clustering. Methods such as region merging and splitting, border smoothing, and noise reduction may be used to increase the accuracy and cohesiveness of the segmented areas.

Image segmentation utilizing clustering techniques such as K-Means may be used in a variety of disciplines such as computer vision, healthcare imaging, object recognition, and analysis of images. It allows for activities including object detection, picture comprehension, image modification, and automatic image-based measurements [120].

4.8.2. Anomaly Detection

Anomaly detection is an important use of clustering techniques that entails recognizing and reporting data points and instances that differ substantially from the usual situation. In clustering, anomaly detection is recognizing datasets or instances which differ considerably from the regular patterns or behavior of a dataset. Clustering methods may be used to find anomalies by taking into account data points that are not associated with any cluster or are positioned distant from the cluster's centroids [121]. Anomaly detection in clustering is the recognition of datasets or instances that deviate significantly from the usual patterns or behavior of a dataset. Clustering algorithms may be used to detect anomalies by considering data points that are not related to any cluster or are located far from the cluster's centroids. The following is an example of how clustering may be used to detect anomalies.

Clustering is useful for detecting anomalies in a variety of disciplines, like identifying fraud, detection of network intrusions, tracking systems, identifying outliers in data from sensors, and finding anomalies for medical care or financial tasks [122]. Anomalies can be found by comparing them with typical patterns in the data using clustering methods, allowing for early detection of anomalous or suspicious cases. In general, using the technique of clustering in the detection of anomalies allows for the recognition of outliers and odd cases, offering significant insights and assisting in the early identification of unusual circumstances across several domains.

4.8.3. Document Clustering

Document clustering is a popular use of clustering techniques that includes organizing an enormous number of documents into useful categories based on textual similarities. It is frequently utilized in a variety of fields including retrieving data, text mining, and handling documents [123]. Identical texts could be clustered together by applying techniques for clustering to documents, allowing for optimal textual data organization, navigating, and evaluation.

Document clustering has various advantages. To begin with, it facilitates retrieving data by grouping papers into clusters, enabling users to rapidly identify relevant materials within specified subjects or themes. It also supports topic modeling, in which clusters indicate unique subjects or themes contained in the document disposal, allowing for a more in-depth analysis of the basic material [124]. The partition clustering technique is widely used in document clustering. It entails grouping comparable publications together based on

their content, allowing for effective textual information organization and retrieval. The K-Medoid approach in document clustering generates typical documents called medoids that act as cluster centroids. Document clustering provides a systematic and understandable organization of textual material by utilizing the K-Medoid clustering method. It enables users to browse and extract insights from vast document repositories more rapidly, allowing for greater efficiency in document management, exploration, and analysis [125].

In the end, document clustering is an effective method for organizing and analyzing vast amounts of textual information. It increases retrieval of data, allows for topic modeling, assists in text categorization, increases handling of documents, and helps in intellectual acquisition. Document clustering enables academics, analysts, and companies to derive relevant insights from massive volumes of textual data by employing clustering techniques.

4.8.4. Social Network Analysis

It is a study of the interactions and patterns that exist inside social networks. The algorithms for clustering are important in social network research because they reveal community frameworks as well as recognize groupings of persons or companies that have similar connection patterns. By utilizing clustering algorithms for social network information, useful knowledge of the network's factors, behavior, and influence may be gleaned [126]. Community identification is an important use of clustering in social network analysis. The goal of community discovery is to find groupings of units in a network that are strongly linked internally and have less connection between them. Clustering techniques, like modularity-based approaches or hierarchical-based clustering, may be utilized to divide the network into coherent groups. Users or institutions inside the same cluster have greater connections and relations, whereas these clusters indicate subgroups and regions within the social network [127].

There are various practical uses for social network clustering. It is employed in systems for recommendations to determine groups of people that share similar tastes or interests, allowing for personalized suggestions. Clustering is additionally utilized in specific advertising, where it aids in the identification of certain customer segments and groups for customized marketing efforts. Furthermore, clustering algorithms help in finding groups of individuals addressing similar themes or engaging in comparable behaviors in the area of internet-based social media analysis [128].

In general, clustering algorithms are important in social network research because they reveal community frameworks, discover groups of people who have comparable connection patterns, and provide information about social interactions, influence, and behavior. Researchers and statisticians can obtain a better knowledge of social networks, targeted specific groups, and examine the development of network frameworks throughout the years by using clustering techniques.

4.8.5. Traffic Analysis

In analyzing traffic, clustering is commonly utilized to obtain information about traffic flow patterns and optimize traffic control systems. Techniques of clustering may extract significant information from data about traffic, resulting in enhanced flow of traffic, congestion management, and transportation strategy [129].

Traffic stream segmentation is one use of clustering in analyzing traffic. Clustering techniques may classify comparable traffic circulation patterns based on parameters like speed, weight, or kind of vehicle. This aids in the identification of various traffic circumstances, like congested sections, open-ended zones, or repeated patterns at specified times of the week. Transportation agencies can design tailored strategies for controlling and optimizing traffic in various regions of the highway network by categorizing traffic flows [130]. Clustering techniques may also be used to investigate traffic behavior and traffic patterns. Transportation managers can acquire data on demand for travel, origin–destination trends, and commuting behavior by grouping comparable trip trajectories or traffic patterns.

These data may be used to build infrastructure, optimize public transit, and develop traffic control techniques.

Clustering is a useful approach in traffic analysis because it provides useful information for the flow of traffic categorization and analyzing journey patterns. Traffic authorities and academics may make educated judgments to boost traffic control, decrease congestion, upgrade the infrastructure for transport, and develop more effective and environmentally friendly modes of transportation by employing clustering algorithms [131].

4.8.6. Customer Segmentation

Customer segmentation is a critical use of clustering techniques in advertising and customer analytics. Companies may use clustering algorithms to divide consumers into various categories on the basis of their similarities, allowing for personalized marketing tactics and focused customer interaction [132].

Clustering techniques are effective tools for customer segmentation because they enable firms to analyze multiple consumer variables and uncover patterns and similarities in the information being analyzed. Businesses may customize promotions, product suggestions, and pricing approaches for every category by categorizing consumers depending on demographics, buying behavior, and inclinations. This improves consumer happiness, retention, and the effectiveness of targeted marketing and loyalty programs. Customer segmentation using clustering also gives valuable information for market analysis and product creation, allowing companies to recognize market preferences, find niche markets, and supply customized solutions [133]. In general, clustering algorithms support customer-centered strategies, maximize the use of resources, and promote business success in cutthroat marketplaces.

4.8.7. Healthcare

Clustering algorithms are useful in healthcare because they provide information about patient groups, illness trends, and medication results. Healthcare practitioners may efficiently analyze and categorize information about patients by using clustering algorithms, resulting in more effective decision making, personalized treatment strategies, and improved medical service. Patient categorization is an important use of clustering in medical applications [134]. Clustering techniques can classify patients depending on their medical records, indications, genetic identities, or therapy effects. This enables healthcare practitioners to identify discrete patient groupings with comparable features, allowing for more targeted treatments and surgeries. Clustering techniques are also used to analyze disease patterns and detect outbreaks. Furthermore, clustering algorithms may be used to allocate and optimize healthcare resources. Clustering techniques can also help discover trends in healthcare utilization, allowing for the detection of locations where medical care could be deficient or overburdened [135].

In general, clustering techniques have enormous possibilities in healthcare, since they will drive advances in patient categorization, analysis of illness patterns, and resource optimization. By utilizing these methodologies, healthcare practitioners may improve personalized treatment and manage illnesses plans and resource allocation, which will eventually contribute to enhanced patient results and the provision of healthcare.

4.8.8. Bioinformatics

Clustering is important in bioinformatics because it contributes to many parts of biological information processing, such as analyzing gene expression and the sequencing of protein clustering. Scholars can find hidden patterns, establish functional linkages, and obtain an understanding of complicated biological processes by applying clustering techniques to biological information [136].

Gene expression evaluation is a key use of clustering in bioinformatics. Clustering techniques are employed to classify genes that have identical expression characteristics among samples and situations. Another key use in bio-informatics is sequenced protein

grouping. Clustering methods are used to classify proteins that have identical sequences and frameworks, hence assisting in protein categorization, functional annotations, and protein-related analyses [136].

In conclusion, clustering techniques have shown to be essential in bio-informatics, allowing analyses of gene expression and sequenced protein clustering. Scholars can reveal patterns, establish links, and gain an understanding of complicated biological processes by clustering biological information. Clustering approaches have considerably increased our knowledge of genes, proteins, and their roles, eventually leading to the improvement of biological studies and scientific discoveries.

4.8.9. Climate Science

Clustering is useful in climate science, as it has the critical capacity to detect basic patterns in complicated climatic datasets. Clustering algorithms are used by climate scientists to classify geographically connected locations with comparable climatic features. Clustering aids in detecting unique zones of climate, trends of temperature variations, and various weather-related phenomena by studying numerous climatic variables such as humidity, temperature, rainfall, and wind patterns. For example, clustering can assist in distinguishing dry, semi-arid, and humid zones by grouping locations that have comparable rainfall patterns [137]. Also, weather conditions are the key factor in agricultural production. As climate change is the primary factor, both components are internally tied to one another in many ways [138,139]. These data are critical for agricultural preparation, handling water resources, as well as disaster preparation. Furthermore, clustering approaches aid in the detection of climatic anomalies that can have a significant impact on worldwide precipitation patterns and regional climate variables.

Depending on the particular purposes and qualities of the data being analyzed, multiple clustering approaches might be applied. K-means clustering is frequently used in climate data analysis to detect different trends. For example, it may be used to combine locations with comparable patterns of precipitation and temperature, assisting in the delineation of climatic zones. Identifying the hierarchy of climatic trends is aided by hierarchical clustering. It aids in the identification of nested climatic areas based on several factors, which is useful for biological and environmental investigations [140,141].

4.8.10. Criminal Profiling

Criminal profiling, also known as behavioral profiling, is an important analytical strategy used to develop cognitive and demographic profiles of unidentified offenders determined by their actions, behaviors, or trends in criminal conduct [142]. Clustering techniques have a major positive impact on criminal profiling, a key component of security agencies. This procedure entails developing criminal profiles based on previous crime information and patterns. Clustering is critical in combining together crime occurrences with similar features, assisting detectives in comprehending modus operandi, locating probable criminals, and forecasting future criminal behavior [143].

The K-means method is a popular clustering method in the identification of criminals. K-means split criminal episodes into unique clusters, each of which represents a particular sort of crime trend. Police departments can gain insights into the similarities and variations between criminal activities by analyzing these clusters. These data not only aid in the creation of criminal profiles but also direct the use of resources and intervention techniques. Moreover, density-based clustering, namely the DBSCAN technique, can help with criminal profiling. DBSCAN finds criminal incidence clusters based on density within a specified geographic area. This method assists in identifying criminal hotspots and trends that may not be as obvious using other methods [144,145].

In general, clustering algorithms provide a data-driven and methodical method for criminal profiling. These approaches enable law enforcement organizations to better comprehend criminal behavior, establish accurate criminal identities, and execute efficient crime prevention or investigative tactics by finding hidden links and combining comparable

instances. Clustering techniques such as K-means and DBSCAN aid in the collection of significant insights from past crime data, hence helping to build precise and efficient criminal profiles. In addition, these profiles enable law enforcement organizations to make more informed judgments, distribute resources more effectively, and improve public safety precautions [146–148].

In conclusion, clustering in climate research enables scientists to decipher the complicated interaction of climatic factors, allowing for precise climate categorization, identifying anomalies, and ecological evaluations. These findings are critical for comprehending our planet's climate change, adopting informed policy choices, and promoting environmentally friendly practices.

Table 6 shows that the above distribution is a generalization that might change based on individual use cases and the standard of the dataset utilized in each area. Furthermore, the ranks (high, medium, low) supplied are relative and reflect the normal predominance of each clustering approach within a certain application area.

Table 6. Overview of the clustering techniques and their application in different domains.

Applications of Clustering	Hierarchical Method	Density-Based Clustering	Grid-Based Clustering	Partition-Based Clustering
Anomaly detection	Medium	Low	Low	High
Customer Segmentation	High	High	Low	High
Bioinformatics	High	High	Low	Low
Healthcare	Medium	Low	Low	High
Traffic Analysis	Low	Low	Medium	High
Social Network Analysis	Low	Medium	Low	High
Document Clustering	Medium	Low	Low	High
Image Segmentation	Low	Low	Medium	High
Climate Science	High	Low	Medium	High
Criminal Profiling	Medium	High	Low	High

5. Discussions

The term "cluster analysis" refers to a group of methods for finding patterns in datasets. Hierarchical clustering, density-based clustering, grid-based clustering, and partition-based clustering are four such methods. While beneficial for detecting outliers and revealing hierarchical linkages, hierarchical clustering is less suited to handling huge datasets. Although density-based clustering is sensitive to parameters and scalability, it is resilient to noise and outliers, making it useful for real-world applications. Large datasets can be handled effectively by grid-based clustering, but complicated clusters provide a challenge. While fast and understandable, partition-based clustering, like k-means, presupposes spherical clusters and calls for predetermined cluster numbers. Each method has its own advantages and disadvantages, making it appropriate for various tasks including fraud detection, image processing, and market analysis. Determining the best approach depends on the data's properties and the analysis's objectives.

In the field of cluster analysis, choosing the right method is essential for obtaining precise and insightful findings. The broad range of clustering algorithms covered above offers a range of complexity, cluster kinds, and performance traits, each suited to certain datasets and analytical objectives. These techniques display a variety of behaviors depending on the size of the dataset, from the exhaustive K-medoid and hierarchical approaches to the sensitive yet efficient K-means variations. K-Medoid and PAM excel at managing random and variable clusters, whereas K-means variations like K-Means++ and K-Mean* improve centroid initialization. However, computational complexity limits the use of hierarchical approaches like agglomerative and divisive hierarchy to smaller datasets and parameter adjustment. The necessity for a complete viewpoint that recognizes the effects

of ignoring hyperparameter tuning within the larger context of AI models is highlighted. The pre-specification of additional hyperparameters may be necessary for some applications in order to obtain better clustering performance. The difficulties of fine-tuning hyperparameters for clustering algorithms are explored in the paper. When taking into account the algorithm's complexity, sensitivity to data cluster features, computational costs, and the trade-off between accuracy and computational cost, these difficulties are made more difficult, also highlighting many cutting-edge hyperparameter tuning techniques. These include the use of metaheuristic algorithms, multi-objective optimization, transfer learning, and automated hyperparameter tweaking through AutoML platforms, as well as sensitivity analysis, which systematically assesses each hyperparameter's impact on clustering outcomes. The authors stress that the quality of clustering results is substantially influenced by the effective selection of hyperparameters, which is in line with the dataset's features and the algorithm's requirements. Finally, clustering approaches provide effective tools for organizing and extracting insights from big and complicated datasets. The most appropriate clustering method is determined by the data properties and the application's unique needs. It is critical to assess the quality of clustering findings in order to assure the validity and use of the produced clusters. Continued study and advancement in clustering algorithms and evaluation approaches will improve data mining's ability to provide meaningful and actionable insights for decision making in a variety of disciplines.

Bias may have been introduced by the criteria used to choose the studies for evaluation. Papers with null or less significant findings were omitted, resulting in an unbalanced depiction of algorithm performance, whereas papers reporting major advancements or unique adaptations of existing algorithms may be more likely to be included. The findings of this study offer a thorough grasp of the advantages and disadvantages of different clustering techniques. This knowledge has applications in algorithm selection, data preparation, and outlier management, as well as regulatory compliance and data privacy policy issues. Furthermore, the future research directions suggested here have the potential to expand cluster analysis, enabling more precise, effective, and understandable clustering solutions for a variety of applications.

6. Conclusions and Future Directions

Unprecedented amounts of data are present in both the commercial and scientific worlds today. Organizations have resorted to data mining, a powerful approach that makes it easier to extract important insights from large and diverse datasets, to navigate this data-rich world efficiently. Data mining involves a wide range of procedures and approaches, such as association rule mining, clustering, classification, and outlier detection. Clustering is a fundamental data mining method that plays an important role in organizing and retrieving meaningful information from enormous datasets. Different clustering algorithms, such as partition-based, hierarchical, grid-based, and density-based techniques, have emerged, each with its own set of pros and cons. The use of a clustering technique is determined by the nature of the data, the intended cluster structure, and the application domain. The partition clustering technique is a popular clustering algorithm that seeks representative items (medoids) to serve as cluster centers. K-Medoid clustering is more resilient to outliers than K-Means clustering.

Other clustering algorithms, such as grid clustering, density-based clustering, and hierarchical clustering, offer alternate options for diverse contexts in addition to partition clustering. Each method has its own set of assumptions and properties that make it ideal for different sorts of data and cluster configurations. A detailed review of the data and the specific aims of the investigation should guide the choice of clustering method. The use of partition clustering assures that the cluster centers align with actual data points, which makes it appropriate for applications requiring interpretability and comprehensibility. Clustering has been used effectively in a variety of disciplines, including anomaly detection, picture segmentation, customer segmentation, and many more. Furthermore, evaluating clustering results is critical for determining the quality and usefulness of clus-

tering algorithms. Various assessment metrics, including entropy, Rand Index, DB Index, and F-measure, can be used to assess the resemblance of clustering findings to ground truth or the internal cohesion and separation of clusters. It is critical to choose appropriate assessment measures that match the analyses.

Finally, clustering approaches provide effective tools for organizing and extracting insights from big and complicated datasets. The most appropriate clustering method is determined by the data properties and the application's unique needs. It is critical to assess the quality of clustering findings in order to assure the validity and use of the produced clusters. Continued study and advancement in clustering algorithms and evaluation approaches will improve data mining's ability to provide meaningful and actionable insights for decision making in a variety of disciplines.

In conclusion, data mining is a powerful technique that allows organizations to extract important insights from big and diverse datasets. It includes a diverse set of approaches and techniques like clustering, classification, association rule mining, and outlier identification. Clustering, in particular, is critical in organizing data items into meaningful groupings, allowing for effective data visualization and information extraction. Different clustering methods, such as partitioned clustering and density-based clustering, have been introduced and used in a variety of fields, like finance, advertising, and medical services. These methods provide several techniques for dealing with data characteristics and cluster formations, allowing organizations to customize the process of clustering.

In the future, data mining and clustering research must concentrate on enhancing the accuracy and efficiency of clustering algorithms. This involves investigating the application of modern technologies such as artificial intelligence and neural networks in clustering, as well as improving the flexibility and productivity of clustering algorithms for analyzing data in real time. Furthermore, the assessment of clustering findings should be developed further to verify the accuracy and use of the produced clusters. To further analyze the quality and usefulness of clustering algorithms in various situations, new assessment metrics and approaches might be investigated. In addition, clustering's applicability goes beyond traditional domains, so there is a need to investigate novel fields where clustering might bring significant insights. For example, anomaly detection and picture segmentation are two particular applications mentioned in the literature, although clustering is likely to be useful in many other fields.

Author Contributions: Conceptualization, M.C. and I.S.; methodology, M.M.; software, E.B.T.; validation, I.A.; formal analysis, M.C. and M.M.; investigation, D.L.R.V. and E.B.T.; data curation, I.S. and M.M.; writing—original draft preparation, M.C. and I.S.; writing—review and editing, M.C. and I.A.; visualization, E.B.T.; supervision, I.A.; project administration, D.L.R.V.; funding acquisition, D.L.R.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the European University of Atlantic.

Data Availability Statement: Not applicable.

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

References

1. Shukor, N.A.; Tasir, Z.; Van der Meijden, H. An examination of online learning effectiveness using data mining. *Procedia-Soc. Behav. Sci.* **2015**, *172*, 555–562. [CrossRef]
2. Schneider, J.; Seidel, S.; Basalla, M.; vom Brocke, J. Reuse, Reduce, Support: Design Principles for Green Data Mining. *Bus. Inf. Syst. Eng.* **2023**, *65*, 65–83. [CrossRef]
3. Ghongade, T.G.; Khobragade, R. Evaluation on Utilization and Emaciation of Data Mining Techniques in Information System. In Proceedings of the 2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON), IEEE, Raigarh, Chhattisgarh, India, 8–10 February 2023; pp. 1–6.
4. Saad, E.; Din, S.; Jamil, R.; Rustam, F.; Mehmood, A.; Ashraf, I.; Choi, G.S. Determining the efficiency of drugs under special conditions from users' reviews on healthcare web forums. *IEEE Access* **2021**, *9*, 85721–85737. [CrossRef]
5. Aslam, S.; Ashraf, I. Data mining algorithms and their applications in education data mining. *Int. J. Adv. Res. Comput. Sci. Manag.* **2014**, *2*, 50–56.

6. Rashid, A.; Asif, S.; Butt, N.A.; Ashraf, I. Feature level opinion mining of educational student feedback data using sequential pattern mining and association rule mining. *Int. J. Comput. Appl.* **2013**, *81*, 31–38. [CrossRef]
7. Rupapara, V.; Rustam, F.; Aljedaani, W.; Shahzad, H.F.; Lee, E.; Ashraf, I. Blood cancer prediction using leukemia microarray gene data and hybrid logistic vector trees model. *Sci. Rep.* **2022**, *12*, 1000. [CrossRef] [PubMed]
8. Indrasiri, P.L.; Lee, E.; Rupapara, V.; Rustam, F.; Ashraf, I. Malicious traffic detection in iot and local networks using stacked ensemble classifier. *Comput. Mater. Contin.* **2022**, *71*, 489–515.
9. Zhou, Z.H. Three perspectives of data mining. *Artif. Intell.* **2003**, *143*, 139–146. [CrossRef]
10. Chen, M.S.; Han, J.; Yu, P.S. Data mining: An overview from a database perspective. *IEEE Trans. Knowl. Data Eng.* **1996**, *8*, 866–883. [CrossRef]
11. Gheware, S.; Kejkar, A.; Tondare, S. Data mining: Task, tools, techniques and applications. *Int. J. Adv. Res. Comput. Commun. Eng.* **2014**, *3*, 8095–8098. [CrossRef]
12. Gupta, M.K.; Chandra, P. A comparative study of clustering algorithms. In Proceedings of the 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, New Delhi, India, 13–15 March 2019; pp. 801–805.
13. Fan, C.Y.; Fan, P.S.; Chan, T.Y.; Chang, S.H. Using hybrid data mining and machine learning clustering analysis to predict the turnover rate for technology professionals. *Expert Syst. Appl.* **2012**, *39*, 8844–8851. [CrossRef]
14. Shaukat, K.; Zaheer, S.; Nawaz, I. Association rule mining: An application perspective. *Int. J. Comput. Sci. Innov.* **2015**, *2015*, 29–38.
15. Muda, Z.; Yassin, W.; Sulaiman, M.N.; Udzir, N.I. Intrusion detection based on k-means clustering and OneR classification. In Proceedings of the 2011 7th International Conference on Information Assurance and Security (IAS), IEEE, Melacca, Malaysia, 5–8 December 2011; pp. 192–197.
16. Kesavaraj, G.; Sukumaran, S. A study on classification techniques in data mining. In Proceedings of the 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), IEEE, Tiruchengode, India, 4–6 July 2013; pp. 1–7.
17. Talagala, P.D.; Hyndman, R.J.; Smith-Miles, K. Anomaly detection in high-dimensional data. *J. Comput. Graph. Stat.* **2021**, *30*, 360–374. [CrossRef]
18. Shu, X.; Ye, Y. Knowledge Discovery: Methods from data mining and machine learning. *Soc. Sci. Res.* **2023**, *110*, 102817. [CrossRef] [PubMed]
19. Oyelade, J.; Isewon, I.; Oladipupo, O.; Emebo, O.; Omogbadegun, Z.; Aromolaran, O.; Uwoghiren, E.; Olaniyan, D.; Olawole, O. Data clustering: Algorithms and its applications. In Proceedings of the 2019 19th International Conference on Computational Science and Its Applications (ICCSA), IEEE, St. Petersburg, Russia, 1–4 July 2019; pp. 71–81.
20. Shafi, I.; Hussain, I.; Ahmad, J.; Kim, P.W.; Choi, G.S.; Ashraf, I.; Din, S. License plate identification and recognition in a non-standard environment using neural pattern matching. *Complex Intell. Syst.* **2022**, *8*, 3627–3639. [CrossRef]
21. Jalal, N.; Mehmood, A.; Choi, G.S.; Ashraf, I. A novel improved random forest for text classification using feature ranking and optimal number of trees. *J. King Saud-Univ.-Comput. Inf. Sci.* **2022**, *34*, 2733–2742. [CrossRef]
22. Ashraf, I.; Hur, S.; Shafiq, M.; Park, Y. Catastrophic factors involved in road accidents: Underlying causes and descriptive analysis. *PLoS ONE* **2019**, *14*, e0223473. [CrossRef]
23. Chakrabarti, D.; Kumar, R.; Tomkins, A. Evolutionary clustering. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; pp. 554–560.
24. Gulati, H.; Singh, P. Clustering techniques in data mining: A comparison. In Proceedings of the 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, New Delhi, India, 11–13 March 2015; pp. 410–415.
25. Jing, W.; Zhao, C.; Jiang, C. An improvement method of DBSCAN algorithm on cloud computing. *Procedia Comput. Sci.* **2019**, *147*, 596–604. [CrossRef]
26. Kang, Z.; Zhao, X.; Peng, C.; Zhu, H.; Zhou, J.T.; Peng, X.; Chen, W.; Xu, Z. Partition level multiview subspace clustering. *Neural Netw.* **2020**, *122*, 279–288. [CrossRef]
27. Mirkin, B. *Clustering: A Data Recovery Approach*; CRC Press: Boca Raton, FL, USA, 2012.
28. Varun, J.; Karthika, R. Achieving Agility in Projects Through Hierarchical Divisive Clustering Algorithm. *J. Electron. Test.* **2022**, *38*, 471–479. [CrossRef]
29. Karypis, G.; Han, E.H.; Kumar, V. Chameleon: Hierarchical clustering using dynamic modeling. *Computer* **1999**, *32*, 68–75. [CrossRef]
30. Gagolewski, M.; Bartoszuk, M.; Cena, A. Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm. *Inf. Sci.* **2016**, *363*, 8–23. [CrossRef]
31. Nielsen, F. *Introduction to HPC with MPI for Data Science*; Springer: Berlin/Heidelberg, Germany, 2016.
32. Celebi, M.E.; Wen, Q.; Hwang, S. An effective real-time color quantization method based on divisive hierarchical clustering. *J. -Real-Time Image Process.* **2015**, *10*, 329–344. [CrossRef]
33. Piccarreta, R.; Billari, F.C. Clustering work and family trajectories by using a divisive algorithm. *J. R. Stat. Soc. Ser. Stat. Soc.* **2007**, *170*, 1061–1078. [CrossRef]
34. Hung, C.C.; Kim, Y. The application of agglomerative clustering in image classification systems. In Proceedings of the IEEE Southeastcon'92, IEEE, Birmingham, AL, USA, 12–15 April 1992; pp. 23–26.

35. Tokuda, E.K.; Comin, C.H.; Costa, L.d.F. Revisiting agglomerative clustering. *Phys. A Stat. Mech. Its Appl.* **2022**, *585*, 126433. [CrossRef]
36. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: An efficient data clustering method for very large databases. *ACM Sigmod Rec.* **1996**, *25*, 103–114. [CrossRef]
37. Lorbeer, B.; Kosareva, A.; Deva, B.; Softić, D.; Ruppel, P.; Küpper, A. Variations on the clustering algorithm BIRCH. *Big Data Res.* **2018**, *11*, 44–53. [CrossRef]
38. Le Quy Nhon, V.; Anh, D.T. A BIRCH-based clustering method for large time series databases. In Proceedings of the New Frontiers in Applied Data Mining: PAKDD 2011 International Workshops, Shenzhen, China, 24–27 May 2011; Revised Selected Papers 15; Springer: Berlin/Heidelberg, Germany, 2012; pp. 148–159.
39. Guha, S.; Rastogi, R.; Shim, K. CURE: An efficient clustering algorithm for large databases. *ACM Sigmod Rec.* **1998**, *27*, 73–84. [CrossRef]
40. Kalnis, P.; Mamoulis, N.; Bakiras, S. On discovering moving clusters in spatio-temporal data. In Proceedings of the Advances in Spatial and Temporal Databases: 9th International Symposium, SSTD 2005, Angra dos Reis, Brazil, 22–24 August 2005; Proceedings 9; Springer: Berlin/Heidelberg, Germany, 2005; pp. 364–381.
41. Safdari-Vaighani, A.; Salehpour, P.; Feizi-Derakhshi, M.R. Detecting Non-Spherical Clusters Using Modified CURE Algorithm. In Proceedings of the 2021 11th International Conference on Computer Engineering and Knowledge (ICCKE), IEEE, Mashhad, Iran, 28–29 October 2021; pp. 369–373.
42. Guha, S.; Rastogi, R.; Shim, K. ROCK: A robust clustering algorithm for categorical attributes. *Inf. Syst.* **2000**, *25*, 345–366. [CrossRef]
43. Almeida, J.; Barbosa, L.; Pais, A.; Formosinho, S. Improving hierarchical cluster analysis: A new method with outlier detection and automatic clustering. *Chemom. Intell. Lab. Syst.* **2007**, *87*, 208–217. [CrossRef]
44. Guo, D.; Zhao, J.; Liu, J. Research and application of improved CHAMELEON algorithm based on condensed hierarchical clustering method. In Proceedings of the 2019 8th International Conference on Networks, Communication and Computing, Luoyang, China, 13–15 December 2019; pp. 14–18.
45. Kriegel, H.P.; Kröger, P.; Sander, J.; Zimek, A. Density-based clustering. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 231–240. [CrossRef]
46. Wang, J.; Zhu, C.; Zhou, Y.; Zhu, X.; Wang, Y.; Zhang, W. From partition-based clustering to density-based clustering: Fast find clusters with diverse shapes and densities in spatial databases. *IEEE Access* **2017**, *6*, 1718–1729. [CrossRef]
47. Khan, K.; Rehman, S.U.; Aziz, K.; Fong, S.; Sarasvady, S. DBSCAN: Past, present and future. In Proceedings of the Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014). IEEE, Bangalore, India, 17–19 February 2014; pp. 232–238.
48. Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Min. Knowl. Discov.* **1998**, *2*, 169–194. [CrossRef]
49. 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.
50. Gialampoukidis, I.; Vrochidis, S.; Kompatsiaris, I. A hybrid framework for news clustering based on the DBSCAN-Martingale and LDA. In Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition, New York, NY, USA, 16–21 July 2016; Springer: Cham, Switzerland, 2016; pp. 170–184.
51. Su, S.; Xiao, L.; Zhang, Z.; Gu, F.; Ruan, L.; Li, S.; He, Z.; Huo, Z.; Yan, B.; Wang, H.; et al. N2DLOF: A new local density-based outlier detection approach for scattered data. In Proceedings of the 2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, Bangkok, Thailand, 18–20 December 2017; pp. 458–465.
52. Rehioui, H.; Idrissi, A.; Abourezq, M.; Zegrari, F. DENCLUE-IM: A new approach for big data clustering. *Procedia Comput. Sci.* **2016**, *83*, 560–567. [CrossRef]
53. Idrissi, A.; Rehioui, H.; Laghrissi, A.; Retal, S. An improvement of DENCLUE algorithm for the data clustering. In Proceedings of the 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA), IEEE, Marrakech, Morocco, 21–23 December 2015; pp. 1–6.
54. Yu, X.G.; Jian, Y. A new clustering algorithm based on KNN and DENCLUE. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, IEEE, Guangzhou, China, 18–21 August 2005; Volume 4, pp. 2033–2038.
55. Ankerst, M.; Breunig, M.M.; Kriegel, H.P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60. [CrossRef]
56. 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]
57. Zhao, Y.; Cao, J.; Zhang, C.; Zhang, S. Enhancing grid-density based clustering for high dimensional data. *J. Syst. Softw.* **2011**, *84*, 1524–1539. [CrossRef]
58. Qiu, B.Z.; Li, X.L.; Shen, J.Y. Grid-based clustering algorithm based on intersecting partition and density estimation. In Proceedings of the Emerging Technologies in Knowledge Discovery and Data Mining: PAKDD 2007 International Workshops Nanjing, China, 22–25 May 2007; Revised Selected Papers 11; Springer: Berlin/Heidelberg, Germany, 2007; pp. 368–377.

59. Bureva, V.; Sotirova, E.; Popov, S.; Mavrov, D.; Traneva, V. Generalized net of cluster analysis process using STING: A statistical information grid approach to spatial data mining. In Proceedings of the Flexible Query Answering Systems: 12th International Conference, FQAS 2017, London, UK, 21–22 June 2017, Proceedings 12; Berlin/Heidelberg, Springer: 2017; pp. 239–248.
60. Lu, Y.; Sun, Y.; Xu, G.; Liu, G. A grid-based clustering algorithm for high-dimensional data streams. In Proceedings of the Advanced Data Mining and Applications: First International Conference, ADMA 2005, Wuhan, China, 22–24 July 2005. Proceedings 1; Springer: Berlin/Heidelberg, Germany, 2005; pp. 824–831.
61. Forster, A.; Murphy, A.L. CLIQUE: Role-free clustering with Q-learning for wireless sensor networks. In Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems, IEEE, Montreal, QC, Canada, 22–26 June 2009; pp. 441–449.
62. Agrawal, R.; Gehrke, J.; Gunopulos, D.; Raghavan, P. Automatic subspace clustering of high dimensional data. *Data Min. Knowl. Discov.* **2005**, *11*, 5–33. [CrossRef]
63. Boonchoo, T.; Ao, X.; Liu, Y.; Zhao, W.; Zhuang, F.; He, Q. Grid-based DBSCAN: Indexing and inference. *Pattern Recognit.* **2019**, *90*, 271–284. [CrossRef]
64. Kellner, D.; Klappstein, J.; Dietmayer, K. Grid-based DBSCAN for clustering extended objects in radar data. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, IEEE, Madrid, Spain, 3–7 June 2012; pp. 365–370.
65. Nazeer, K.A.; Kumar, S.M.; Sebastian, M. Enhancing the k-means clustering algorithm by using a O (n logn) heuristic method for finding better initial centroids. In Proceedings of the 2011 Second International Conference on Emerging Applications of Information Technology, IEEE, Kolkata, India, 19–20 February 2011; pp. 261–264.
66. Na, S.; Xumin, L.; Yong, G. Research on k-means clustering algorithm: An improved k-means clustering algorithm. In Proceedings of the 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, IEEE, Jian, China, 2–4 April 2010; pp. 63–67.
67. Ren, S.; Fan, A. K-means clustering algorithm based on coefficient of variation. In Proceedings of the 2011 4th International Congress on Image and Signal Processing, IEEE, Shanghai, China, 15–17 October 2011; Volume 4, pp. 2076–2079.
68. Lin, K.; Li, X.; Zhang, Z.; Chen, J. A K-means clustering with optimized initial center based on Hadoop platform. In Proceedings of the 2014 9th International Conference on Computer Science & Education, IEEE, Vancouver, BC, Canada, 22–24 August 2014; pp. 263–266.
69. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm, Computers & Geosciences. *Volume* **1984**, *10*, 2–3.
70. Lei, T.; Jia, X.; Zhang, Y.; Liu, S.; Meng, H.; Nandi, A.K. Superpixel-based fast fuzzy C-means clustering for color image segmentation. *IEEE Trans. Fuzzy Syst.* **2018**, *27*, 1753–1766. [CrossRef]
71. Velmurugan, T. Performance based analysis between k-Means and Fuzzy C-Means clustering algorithms for connection oriented telecommunication data. *Appl. Soft Comput.* **2014**, *19*, 134–146.
72. Banerjee, S.; Choudhary, A.; Pal, S. Empirical evaluation of k-means, bisecting k-means, fuzzy c-means and genetic k-means clustering algorithms. In Proceedings of the 2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), IEEE, Dhaka, Bangladesh, 19–20 December 2015; pp. 168–172.
73. Kannan, S.; Ramathilagam, S.; Sathya, A. Robust fuzzy C-means in classifying breast tissue regions. In Proceedings of the 2009 International Conference on Advances in Recent Technologies in Communication and Computing, IEEE, Kottayam, India, 27–28 October 2009; pp. 543–545.
74. Van Lung, H.; Kim, J.M. A generalized spatial fuzzy c-means algorithm for medical image segmentation. In Proceedings of the 2009 IEEE International Conference on Fuzzy Systems, IEEE, Jeju, Republic of Korea, 20–24 August 2009; pp. 409–414.
75. Zhou, H.; Schaefer, G.; Sadka, A.H.; Celebi, M.E. Anisotropic mean shift based fuzzy c-means segmentation of dermoscopy images. *IEEE J. Sel. Top. Signal Process.* **2009**, *3*, 26–34. [CrossRef]
76. Agarwal, S.; Yadav, S.; Singh, K. Notice of Violation of IEEE Publication Principles: K-means versus k-means++ clustering technique. In Proceedings of the 2012 Students Conference on Engineering and Systems, IEEE, Allahabad, India, 16–18 March 2012; pp. 1–6.
77. Aggarwal, S.; Singh, P. Cuckoo, Bat and Krill Herd based k-means++ clustering algorithms. *Clust. Comput.* **2019**, *22*, 14169–14180. [CrossRef]
78. Gao, M.; Pan, S.; Chen, S.; Li, Y.; Pan, N.; Pan, D.; Shen, X. Identification method of electrical load for electrical appliances based on K-Means++ and GCN. *IEEE Access* **2021**, *9*, 27026–27037. [CrossRef]
79. Zhang, M.; Duan, K.-F. Improved research to K-means initial cluster centers. In Proceedings of the 2015 Ninth International Conference on Frontier of Computer Science and Technology, IEEE, Dalian, China, 26–28 August 2015; pp. 349–353.
80. Tzortzis, G.; Likas, A. The MinMax k-Means clustering algorithm. *Pattern Recognit.* **2014**, *47*, 2505–2516. [CrossRef]
81. Hung, M.C.; Wu, J.; Chang, J.H.; Yang, D.L. An Efficient k-Means Clustering Algorithm Using Simple Partitioning. *J. Inf. Sci. Eng.* **2005**, *21*, 1157–1177.
82. Bansal, A.; Sharma, M.; Goel, S. Improved k-mean clustering algorithm for prediction analysis using classification technique in data mining. *Int. J. Comput. Appl.* **2017**, *157*, 0975–8887. [CrossRef]
83. Swarndeep Saket, J.; Pandya, S. An overview of partitioning algorithms in clustering techniques. *Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET)* **2016**, *5*, 1943–1946.

84. Madhulatha, T.S. Comparison between k-means and k-medoids clustering algorithms. In Proceedings of the International Conference on Advances in Computing and Information Technology, Chennai, India, 15–17 July 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 472–481.
85. Surya, P.; Laurence Aroquiaraj, I. Performance analysis of K-means and K-medoid clustering algorithms using agriculture dataset. *J. Emerg. Technol. Innov. Res. (JETIR)* **2019**, *6*, 539–545.
86. Chitrakar, R.; Huang, C. Anomaly based intrusion detection using hybrid learning approach of combining k-medoids clustering and naive bayes classification. In Proceedings of the 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, Shanghai, China, 21–23 September 2012; pp. 1–5.
87. Zhang, H.; Cheng, N.; Zhang, Y.; Li, Z. Label flipping attacks against Naive Bayes on spam filtering systems. *Appl. Intell.* **2021**, *51*, 4503–4514. [CrossRef]
88. Rdusseeun, L.; Kaufman, P. Clustering by means of medoids. In Proceedings of the Statistical Data Analysis Based on the L1 Norm Conference, Neuchatel, Switzerland, 31 August–4 September 1987; Volume 31.
89. Kariv, O.; Hakimi, S.L. An algorithmic approach to network location problems. I: The p-centers. *SIAM J. Appl. Math.* **1979**, *37*, 513–538. [CrossRef]
90. Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
91. Li, Z.; Wang, G.; He, G. Milling tool wear state recognition based on partitioning around medoids (PAM) clustering. *Int. J. Adv. Manuf. Technol.* **2017**, *88*, 1203–1213. [CrossRef]
92. Song, H.; Lee, J.G.; Han, W.S. PAMAE: Parallel k-medoids clustering with high accuracy and efficiency. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, alifax, NS, Canada, 13–17 August 2017; pp. 1087–1096.
93. Yin, J.; Zhou, D.; Xie, Q.Q. A clustering algorithm for time series data. In Proceedings of the 2006 Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06). IEEE, Taipei, China, 4–7 December 2006; pp. 119–122.
94. Renjith, S.; Sreekumar, A.; Jathavedan, M. Performance evaluation of clustering algorithms for varying cardinality and dimensionality of data sets. *Mater. Today Proc.* **2020**, *27*, 627–633. [CrossRef]
95. Ng, R.T.; Han, J. CLARANS: A method for clustering objects for spatial data mining. *IEEE Trans. Knowl. Data Eng.* **2002**, *14*, 1003–1016. [CrossRef]
96. Schubert, E.; Rousseeuw, P.J. Fast and eager k-medoids clustering: O (k) runtime improvement of the PAM, CLARA, and CLARANS algorithms. *Inf. Syst.* **2021**, *101*, 101804. [CrossRef]
97. Wei, C.P.; Lee, Y.H.; Hsu, C.M. Empirical comparison of fast clustering algorithms for large data sets. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, IEEE, Maui, HI, USA, 7 January 2000; pp. 1–10.
98. Bischl, B.; Binder, M.; Lang, M.; Pielok, T.; Richter, J.; Coors, S.; Thomas, J.; Ullmann, T.; Becker, M.; Boulesteix, A.L.; et al. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2023**, *13*, e1484. [CrossRef]
99. Liu, X. Hyperparameter-free localized simple multiple kernel K-means with global optimum. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 8566–8576. [CrossRef]
100. Calik, N.; Güneş, F.; Koziel, S.; Pietrenko-Dabrowska, A.; Belen, M.A.; Mahouti, P. Deep-learning-based precise characterization of microwave transistors using fully-automated regression surrogates. *Sci. Rep.* **2023**, *13*, 1445. [CrossRef]
101. Karaman, A.; Karaboga, D.; Pacal, I.; Akay, B.; Basturk, A.; Nalbantoglu, U.; Coskun, S.; Sahin, O. Hyper-parameter optimization of deep learning architectures using artificial bee colony (ABC) algorithm for high performance real-time automatic colorectal cancer (CRC) polyp detection. *Appl. Intell.* **2023**, *53*, 15603–15620. [CrossRef]
102. Thielmann, A.; Weisser, C.; Kneib, T.; Säfken, B. Coherence based document clustering. In Proceedings of the 2023 IEEE 17th International Conference on Semantic Computing (ICSC), IEEE, Laguna Hills, CA, USA, 1–3 February 2023; pp. 9–16.
103. Vinh, N.X.; Epps, J. A novel approach for automatic number of clusters detection in microarray data based on consensus clustering. In Proceedings of the 2009 Ninth IEEE International Conference on Bioinformatics and BioEngineering, IEEE, Taichung, Taiwan, 22–24 June 2009; pp. 84–91.
104. Abuobieda, A.; Salim, N.; Binwahlan, M.S.; Osman, A.H. Differential evolution cluster-based text summarization methods. In Proceedings of the 2013 International Conference on Computing, Electrical and Electronic Engineering (ICCEEE), IEEE, Khartoum, Sudan, 26–28 August 2013; pp. 244–248.
105. Gavioli, A.; de Souza, E.G.; Bazzi, C.L.; Schenatto, K.; Betzek, N.M. Identification of management zones in precision agriculture: An evaluation of alternative cluster analysis methods. *Biosyst. Eng.* **2019**, *181*, 86–102. [CrossRef]
106. Jiang, H.; Yi, S.; Li, J.; Yang, F.; Hu, X. Ant clustering algorithm with K-harmonic means clustering. *Expert Syst. Appl.* **2010**, *37*, 8679–8684. [CrossRef]
107. Campello, R.J. A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. *Pattern Recognit. Lett.* **2007**, *28*, 833–841. [CrossRef]
108. Tambunan, H.B.; Barus, D.H.; Hartono, J.; Alam, A.S.; Nugraha, D.A.; Usman, H.H.H. Electrical peak load clustering analysis using K-means algorithm and silhouette coefficient. In Proceedings of the 2020 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP), IEEE, Bandung, Indonesia, 23–24 September 2020; pp. 258–262.

109. Kathuria, A.; Mukhopadhyay, D.; Thakur, N. Evaluating cohesion score with email clustering. In Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019), Chandigarh, India, 12–13 October 2019; Springer: Singapore, 2019; pp. 107–119.
110. Ncir, C.E.B.; Hamza, A.; Bouaguel, W. Parallel and scalable Dunn Index for the validation of big data clusters. *Parallel Comput.* **2021**, *102*, 102751. [CrossRef]
111. Wu, S.; Chow, T.W. Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density. *Pattern Recognit.* **2004**, *37*, 175–188. [CrossRef]
112. Zhou, S.; Xu, Z. A novel internal validity index based on the cluster centre and the nearest neighbour cluster. *Appl. Soft Comput.* **2018**, *71*, 78–88. [CrossRef]
113. Li, K.; Cao, X.; Ge, X.; Wang, F.; Lu, X.; Shi, M.; Yin, R.; Mi, Z.; Chang, S. Meta-heuristic optimization-based two-stage residential load pattern clustering approach considering intra-cluster compactness and inter-cluster separation. *IEEE Trans. Ind. Appl.* **2020**, *56*, 3375–3384.
114. Łukasik, S.; Kowalski, P.A.; Charytanowicz, M.; Kulczycki, P. Clustering using flower pollination algorithm and Calinski-Harabasz index. In Proceedings of the 2016 IEEE congress on evolutionary computation (CEC), IEEE, Vancouver, BC, Canada, 24–29 July 2016; pp. 2724–2728.
115. Ansari, Z.; Azeem, M.; Ahmed, W.; Babu, A.V. Quantitative evaluation of performance and validity indices for clustering the web navigational sessions. *arXiv* **2015**, arXiv:1507.03340.
116. Zhao, H.; Liang, J.; Hu, H. Clustering validity based on the improved hubert\gamma statistic and the separation of clusters. In Proceedings of the First International Conference on Innovative Computing, Information and Control-Volume I (ICICIC'06), IEEE, Beijing, China, 30 August–1 September 2006; Volume 2, pp. 539–543.
117. Yaslan, Y.; Cataltepe, Z. A Comparison Framework of Similarity Metrics Used for Web Access Log Analysis. In Proceedings of the MLDM Posters, Leipzig, Germany, 18–20 July 2007; pp. 144–152.
118. Sriwastwa, A.; Prakash, S.; Rana, M.; Swarit, S.; Kumari, K.; Sahu, S.S. Detection of pests using color based image segmentation. In Proceedings of the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), IEEE, Coimbatore, India, 20–21 April 2018; pp. 1393–1396.
119. Nguyen, T.T.; Krishnakumari, P.; Calvert, S.C.; Vu, H.L.; Van Lint, H. Feature extraction and clustering analysis of highway congestion. *Transp. Res. Part Emerg. Technol.* **2019**, *100*, 238–258. [CrossRef]
120. Jiang, Y.; Gu, X.; Wu, D.; Hang, W.; Xue, J.; Qiu, S.; Lin, C.T. A novel negative-transfer-resistant fuzzy clustering model with a shared cross-domain transfer latent space and its application to brain CT image segmentation. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2020**, *18*, 40–52. [CrossRef]
121. Li, J.; Izakian, H.; Pedrycz, W.; Jamal, I. Clustering-based anomaly detection in multivariate time series data. *Appl. Soft Comput.* **2021**, *100*, 106919. [CrossRef]
122. Ariyaluran Habeeb, R.A.; Nasaruddin, F.; Gani, A.; Amanullah, M.A.; Abaker Targio Hashem, I.; Ahmed, E.; Imran, M. Clustering-based real-time anomaly detection—A breakthrough in big data technologies. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3647. [CrossRef]
123. Janani, R.; Vijayarani, S. Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Syst. Appl.* **2019**, *134*, 192–200. [CrossRef]
124. Bafna, P.; Pramod, D.; Vaidya, A. Document clustering: TF-IDF approach. In Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), IEEE, Chennai, India, 3–5 March 2016; pp. 61–66.
125. Abualigah, L.M.; Khader, A.T.; Hanandeh, E.S. A new feature selection method to improve the document clustering using particle swarm optimization algorithm. *J. Comput. Sci.* **2018**, *25*, 456–466. [CrossRef]
126. Alsayat, A.; El-Sayed, H. Social media analysis using optimized K-Means clustering. In Proceedings of the 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA), IEEE, Towson, MD, USA, 8–10 June 2016; pp. 61–66.
127. Li, P.; Dau, H.; Puleo, G.; Milenkovic, O. Motif clustering and overlapping clustering for social network analysis. In Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
128. Mishra, N.; Schreiber, R.; Stanton, I.; Tarjan, R.E. Clustering social networks. In Proceedings of the International Workshop on Algorithms and Models for the Web-Graph, San Diego, CA, USA, 11–12 December 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 56–67.
129. Liu, Y.; Li, W.; Li, Y. Network traffic classification using k-means clustering. In Proceedings of the Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007), IEEE, Iowa City, IA, USA, 13–15 August 2007; pp. 360–365.
130. Toshniwal, D.; Chaturvedi, N.; Parida, M.; Garg, A.; Choudhary, C.; Choudhary, Y. Application of clustering algorithms for spatio-temporal analysis of urban traffic data. *Transp. Res. Procedia* **2020**, *48*, 1046–1059. [CrossRef]
131. Erman, J.; Arlitt, M.; Mahanti, A. Traffic classification using clustering algorithms. In Proceedings of the 2006 SIGCOMM workshop on Mining network data, Pisa, Italy, 11–15 September 2006; pp. 281–286.
132. Hung, P.D.; Lien, N.T.T.; Ngoc, N.D. Customer segmentation using hierarchical agglomerative clustering. In Proceedings of the 2nd International Conference on Information Science and Systems, Tokyo, Japan, 16–19 March 2019; pp. 33–37.

133. Lefait, G.; Kechadi, T. Customer segmentation architecture based on clustering techniques. In Proceedings of the 2010 Fourth International Conference on Digital Society, IEEE, Saint Maarten, Netherlands Antilles, 10–16 February 2010; pp. 243–248.
134. Hillerman, T.; Souza, J.C.F.; Reis, A.C.B.; Carvalho, R.N. Applying clustering and AHP methods for evaluating suspect healthcare claims. *J. Comput. Sci.* **2017**, *19*, 97–111. [CrossRef]
135. Paul, R.; Hoque, A.S.M.L. Clustering medical data to predict the likelihood of diseases. In Proceedings of the 2010 fifth international conference on digital information management (ICDIM), IEEE, Thunder Bay, ON, Canada, 5–8 July 2010; pp. 44–49.
136. Tasoulis, D.; Plagianakos, V.; Vrahatis, M. Unsupervised clustering of bioinformatics data. In Proceedings of the European Symposium on Intelligent Technologies, Hybrid Systems and Their Implementation on Smart Adaptive Systems, Eunite, Aachen, Germany, 10–12 June 2004; pp. 47–53.
137. Bochenek, B.; Ustrnul, Z. Machine learning in weather prediction and climate analyses—Applications and perspectives. *Atmosphere* **2022**, *13*, 180. [CrossRef]
138. Singh, S.; Babu, K.S.; Singh, S. Machine learning approach for climate change impact assessment in agricultural production. In *Visualization Techniques for Climate Change with Machine Learning and Artificial Intelligence*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 317–335.
139. Nguyen, T.T.; Grote, U.; Neubacher, F.; Rahut, D.B.; Do, M.H.; Paudel, G.P. Security risks from climate change and environmental degradation: implications for sustainable land use transformation in the Global South. *Curr. Opin. Environ. Sustain.* **2023**, *63*, 101322. [CrossRef]
140. Sadeghi, M.; Naghedi, R.; Behzadian, K.; Shamshirgaran, A.; Tabrizi, M.R.; Maknoon, R. Customisation of green buildings assessment tools based on climatic zoning and experts judgement using K-means clustering and fuzzy AHP. *Build. Environ.* **2022**, *223*, 109473. [CrossRef]
141. Fahad, S.; Su, F.; Khan, S.U.; Naeem, M.R.; Wei, K. Implementing a novel deep learning technique for rainfall forecasting via climatic variables: An approach via hierarchical clustering analysis. *Sci. Total Environ.* **2023**, *854*, 158760. [CrossRef]
142. Vukovic, I. Truth-value unconstrained face clustering for identity resolution in a distributed environment of criminal police information systems. *Eng. Appl. Artif. Intell.* **2023**, *124*, 106576. [CrossRef]
143. Kuppala, J.; Srinivas, K.K.; Anudeep, P.; Kumar, R.S.; Vardhini, P.H. Benefits of Artificial Intelligence in the Legal System and Law Enforcement. In Proceedings of the 2022 International Mobile and Embedded Technology Conference (MECON), IEEE, Noida, India, 10–11 March 2022; pp. 221–225.
144. Al-Ghushami, A.H.; Syed, D.; Sessa, J.; Zainab, A. Intelligent Automation of Crime Prediction using Data Mining. In Proceedings of the 2022 IEEE 31st International Symposium on Industrial Electronics (ISIE), IEEE, Anchorage, AK, USA, 1–3 June 2022; pp. 245–252.
145. Garcia-Zanabria, G.; Raimundo, M.M.; Poco, J.; Nery, M.B.; Silva, C.T.; Adorno, S.; Nonato, L.G. Cripav: Street-level crime patterns analysis and visualization. *IEEE Trans. Vis. Comput. Graph.* **2021**, *28*, 4000–4015. [CrossRef] [PubMed]
146. Zhou, Y.; Wang, F.; Zhou, S. The Spatial Patterns of the Crime Rate in London and Its Socio-Economic Influence Factors. *Soc. Sci.* **2023**, *12*, 340. [CrossRef]
147. William, P.; Shrivastava, A.; Shunmuga Karpagam, N.; Mohanaprakash, T.; Tongkachok, K.; Kumar, K. Crime analysis using computer vision approach with machine learning. In *Mobile Radio Communications and 5G Networks: Proceedings of Third MRCN 2022*; Springer: Singapore, 2023; pp. 297–315.
148. Jayapratha, C.; Chitra, H.S.H.; Priya, R.M. Suspicious Crime Identification and Detection Based on Social Media Crime Analysis Using Machine Learning Algorithms. In *Soft Computing: Theories and Applications: Proceedings of SoCTA 2022*; Springer: Singapore, 2023; pp. 831–843.