

Clustering algorithms: taxonomies, comparative studies, and emerging approaches based on meta-features of datasets

Yesenia Santana-Cardoso^{1,2}, Alicia Martínez-Rebollar¹, Eddie Clemente¹, Carlos Fernando Moreno-Calderón¹, Dante Mújica-Vargas¹

¹National Technological Institute of Mexico (TecNM) / National Center for Research and Technological Development (CENIDET), Cuernavaca, Morelos, México

²Technological Subsystem University / Technological University of the Northern Region of Guerrero, Iguala Guerrero, México

ARTICLE INFO	ABSTRACT
Published Online: 29 November 2025 Corresponding Author: Alicia Martinez-Rebollar KEYWORDS: Clustering Algorithms, Taxonomies, Comparative Studies, Dataset Meta-Features, Unsupervised Learning.	Clustering is a fundamental technique in Data Science used to identify hidden structures and patterns in unlabeled datasets. However, the wide variety of algorithmic families and the heterogeneity of real-world data make the selection of an appropriate clustering algorithm a persistent challenge. This article presents a literature review of clustering algorithms, considering various taxonomies proposed in previous studies and highlighting emerging trends based on dataset meta-features. Our review addresses the main families of clustering algorithms: partitioning, hierarchical, density-based, model-based, and grid-based emphasizing their underlying assumptions, computational properties, and practical advantages. Additionally, comparative studies published over the past two decades are analyzed to identify performance patterns, recurrent limitations, and the scenarios in which each technique tends to perform best. Finally, this article examines contemporary approaches that incorporate meta-features, meta-learning strategies, and algorithm recommendation models aimed at reducing the traditional trial-and-error process through systematic mechanisms that link dataset profiles with suitable algorithms. This state-of-the-art review provides an updated and structured overview of the field, highlighting opportunities to advance toward intelligent clustering algorithm selection.

I. INTRODUCTION

Data clustering is one of the most widely used techniques in Data Science and Machine Learning, as it enables the discovery of internal structure, hidden relationships, and natural groupings within unlabeled datasets [1]. Its applications span numerous domains, including health sciences, engineering, environmental studies, social sciences, marketing, and cybersecurity, where understanding data organization is essential for decision-making, pattern identification, and exploratory analysis [2], [3]. Despite its broad applicability, selecting the most appropriate clustering algorithm remains a considerable challenge due to the diversity of algorithmic families and the inherent heterogeneity of real-world data. Figure 1 shows the conceptual relationship among three fundamental elements: Data Science, machine learning algorithms, and algorithm implementation. At this intersection lie clustering algorithms, which integrate both theoretical foundations and practical considerations for their application in different contexts.

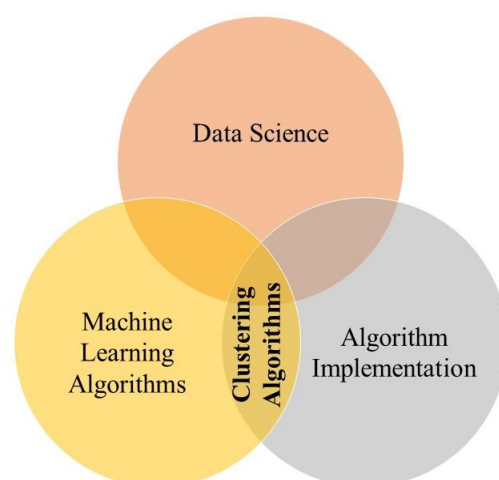


Figure 1: Relationship among Data Science, Machine Learning, and Algorithm Implementation.

Over the past decades, a wide variety of clustering methods have been proposed, ranging from classical techniques such as *k-means* and hierarchical clustering to advanced

approaches based on density, probabilistic models, optimization, or grid structures [4]. These algorithms differ not only in their methodological foundations but also in their assumptions, computational behavior, sensitivity to noise, parameter requirements, and suitability for different types of data. As a result, no clustering method consistently outperforms the others across all scenarios.

To address this diversity, several taxonomies [5] have been developed that provide conceptual frameworks for organizing clustering algorithms according to their objectives, mechanisms, or structural assumptions. Likewise, multiple comparative studies have evaluated their performance across different datasets, metrics, and scenarios. Although these works contribute valuable insights, their conclusions often depend on the dataset used and rarely generalize to broader contexts. This reinforces the need for systematic approaches that support algorithm selection based on measurable properties of the data.

In recent years, research lines based on data meta-characteristics, meta-learning strategies, and algorithm recommendation models have emerged. Meta-characteristics, such as statistical indicators, structural descriptors, geometric properties, and complexity measures, enable the quantitative characterization of datasets and the linkage of these descriptions to the suitability of certain algorithms. These emerging approaches aim to reduce the traditional trial-and-error practice, moving toward more intelligent and data-driven decision-making in unsupervised learning.

In this context, the present article provides a structured review of clustering algorithms, with emphasis on their taxonomies, the comparative analyses reported in the literature, and recent approaches that incorporate dataset meta-characteristics. The objective is to identify persistent challenges and highlight opportunities to advance toward a more systematic and intelligent selection of clustering algorithms.

This article is organized as follows: Section II presents the theoretical foundations and state of the art; Section III details a comparative analysis of the algorithms examined. Section IV presents the reference datasets; Section V discusses several clustering applications; and finally, Section VI provides the conclusions and future work.

II. THEORETICAL FOUNDATIONS AND STATE OF THE ART

This section presents the state of the art, synthesizing the most relevant and recent contributions in the specialized literature. It outlines current advances, methodological approaches, and emerging trends that underpin the theoretical framework and delineate the principal research directions associated with the problem under study.

A clustering algorithm is an unsupervised machine learning technique that organizes a dataset into groups of elements that are similar to one another. Its application is

relevant in contexts where high dimensionality, heterogeneity, and the absence of prior knowledge make it difficult to understand how data are internally structured. Each clustering algorithm has its own advantages and disadvantages, and selecting the appropriate algorithm depends on several factors, such as the type of data, dataset size, data dimensionality, data distribution, result interpretability, and the objectives of the analysis. The main families of clustering algorithms are outlined below, each characterized by distinct methodological foundations and specific approaches to data organization.

A. *partitional algorithms*

The *k-means* algorithm [6] is a well-known clustering technique in the field of unsupervised learning. It consists of partitioning a set of n objects into $k \geq 2$ groups, such that the objects belonging to the same group share similar characteristics, while differing from those in other groups.

There are various research studies that improve existing clustering methods. An example is the work described in [7], which proposes a new convergence condition consisting of stopping the *K-means* algorithm when a local optimum is reached or when no further changes of objects between groups occur.

In the research described in [8], a criterion is proposed to balance processing time and solution quality in the *K-means* algorithm. The authors implement an improvement in the algorithm's convergence phase using the Pareto principle. They evaluate the enhanced algorithm on five Big Data type datasets, significantly reducing processing time and improving solution quality.

The *K-means* algorithm has been widely used in the Big Data domain because large instances are difficult to cluster. The research described in [9] addresses this problem by improving the *K-means* algorithm for clustering Big Data type datasets. The authors developed a heuristic called Honeycomb (HC), which is based on the number of dimensions and the number of centroids that form the neighborhood, in order to reduce the number of distance calculations between objects and cluster centers. This approach achieves up to a 90% reduction in processing time and less than a 1% reduction in solution quality.

There are different versions and improvements of the K-Means algorithm. In the study presented in [10], an exploratory analysis is described regarding the behavior of the main variants of the K-Means algorithm (Hartigan–Wong, Lloyd, Forgy, and MacQueen) when solving some of the difficult instance sets from the Fundamental Clustering Problems Suite (FCPS), a benchmark reference. These variants are implemented in the R language and allow computing the minimum and maximum intra-cluster distances of the final clustering. Below is the mathematical formulation of the K-Means algorithm.

Let $N = \{x_i, \dots, x_n\}$ be the dataset to be partitioned, where $x_i \in \mathbb{R}^d$ for $i = 1, \dots, n$ y $d \geq 1$ is the number of dimensions.

$$P: \text{minimize } J(W, M) = \sum_{i=1}^n \sum_{j=1}^k w_{ij} d(x_i, \mu_j)$$

Subject to $\sum_{j=1}^k w_{ij} = 1, i = 1, \dots, n$,
 $w_{ij} = 0 \text{ or } 1, i = 1, \dots, n, j = 1, \dots, k$,
 where $w_{ij} = 1$ if and only if the object x_i belongs to cluster μ_j y $d(x_i, \mu_j)$ denotes the Euclidean distance between x_i y μ_j , for $i = 1, \dots, n, j = 1, \dots, k$.

The *K-means* algorithm consists of four stages: initialization, classification, centroid computation, and convergence.

The *Fuzzy C-Means* algorithm [11] is also recognized for its data partitioning, similar to *K-means*. However, its fuzzy partition allows each object to belong to multiple groups with different degrees of membership.

There are research studies focused on combining algorithms to improve results. The study presented in [12] proposes a hybrid variant of the Fuzzy C-Means and K-Means algorithms to address large datasets such as those found in Big Data, introducing a new approach called Hybrid OK-Means Fuzzy C-Means (HOFCM), which optimizes the parameter values of the membership matrix.

A different improvement of the Fuzzy C-Means algorithm is described in [13]. The authors use the K-Means algorithm to generate the final centroids, which, through a transformation process known as One-Hot encoding, produce the membership matrix required by the Fuzzy C-Means algorithm. To obtain this encoding, the study proposes selecting attributes using entropy weighting and the Box–Cox transformation. This work highlights improvements in solution time, achieving a 5.19% gain in the best case, and a 5.56% improvement in the silhouette coefficient compared to the standard Fuzzy C-Means algorithm.

Another application involving heuristics is described in [14], where the authors implement the Particle Swarm Optimization algorithm to select the optimal initial centroids for Fuzzy C-Means. Its performance is evaluated using three metrics Rand Index, F-measure, and Objective Function through which the authors determine that the proposed method converges in less time than the conventional algorithm. Their results highlight an improvement of 0.5% in the objective function value compared to the standard Fuzzy C-Means algorithm.

Other methods for improving cluster initialization involve analyzing data density. An example is the study described in [15], in which the authors propose using the Density Peaks Clustering (DPC) algorithm to identify the initial centroids and the groups with which the dataset will be processed. Among the reported results, they describe an accuracy of 84.99% on the dataset known as *Jain* and an adjusted Rand index value of 48.48%, both of which outperform other implemented algorithms.

Below is the mathematical formulation of the Fuzzy C-Means algorithm.

Let $N = \{x_1, \dots, x_n\}$ be the dataset to be partitioned, where $x_i \in \mathbb{R}^d$ for $i = 1, \dots, n$ y c is the number of groups where $2 \leq c \leq n$.

$$J_m(U, V) = \sum_{i=1}^n \sum_{j=1}^c \mu_{ij}^m \|x_i - v_j\|_2^2 \quad (2)$$

where $U = \mu_{ij}$ is the degree of membership of each object i to each group j ; $V = \{v_1, \dots, v_n\}$ is the set of centroids, where v_j is the centroid of group j ; m is the weighting exponent, which modifies the degree of membership, $m > 1$ y $\|x_i - v_j\|_2^2$ denotes the distance calculations from objects to centroids using the Euclidean norm. The optimal clustering of the data X as pairs (U, V) that locally minimize J_m yields the following expressions.

$$\mu_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{m}{m-1}}} \quad \begin{matrix} 1 \leq i \leq n; \\ 1 \leq j \leq c \end{matrix} \quad (3)$$

$$v_j = \frac{\sum_{i=1}^n (\mu_{ij})^m x_i}{\sum_{i=1}^n (\mu_{ij})^m}; \quad 1 \leq j \leq c \quad (4)$$

where $x_i = (x_1, x_2, \dots, x_d)$ and $v_j = (v_1, v_2, \dots, v_d)$ are vectors that belong to the space $\mathbb{R}^d$.

B. Hierarchy-Based Algorithms

Hierarchical-based algorithms construct groups through successive merging. A popular example within this category is the Agglomerative Hierarchical Clustering algorithm [16]. Unlike partitional methods, this algorithm does not require specifying the initial number of clusters; instead, it generates a hierarchical structure called a dendrogram.

One application of this algorithm is the integration of data obtained from wireless sensors. The study in [17] describes the application of the Agglomerative Hierarchical Clustering algorithm in clustering data extracted from interconnected sensors, addressing the problem of node failures during data transmission within the sensor network. The authors implement this algorithm in two phases: backbone formation and recovery process. In the first phase, the algorithm is applied to determine the cluster centers of the networks, and in the second phase, mobile nodes are deployed at the cluster centers, achieving improved efficiency in data traffic across the nodes. Below is the mathematical formulation of the HAC algorithm.

Let $N = \{x_1, \dots, x_n\}$ be the dataset to be partitioned, where $x_i \in \mathbb{R}^d$ for $i = 1, \dots, n$ and each object forms an individual group $C^{(0)} = \{x_1, x_2, \dots, x_n\}$, the distance between objects is computed $D(i, j) = \|x_i - x_j\|_2$.

$$\Delta(A, B)$$

where A and B are two clusters, and the linkage criterion is defined.

$$D_{min}(C_i, C_j) = \min_{x \in C_i, y \in C_j} \|x - y\|_2 \quad (5)$$

$$D_{max}(C_i, C_j) = \max_{x \in C_i, y \in C_j} \|x - y\|_2 \quad (6)$$

$$D_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} \|x - y\|_2 \quad (7)$$

$$D_{ward}(C_i, C_j) = \frac{|C_i||C_j|}{|C_i| + |C_j|} \|v_i - v_j\|_2^2 \quad (8)$$

where v_i y v_j are the centroids C_i and C_j respectively.

C. Density-Based Algorithms

Density-based algorithms create clusters by identifying regions in the n -dimensional space where dense groups of points exist. One of the most widely used algorithms in this category is DBSCAN [18], which identifies dense groups of points and separates them from sparse regions; it also detects outliers in the point distribution. Some studies focus on the initial parameter values of the DBSCAN algorithm to improve its results. An example is the work in [19], where this problem is addressed using the KNN algorithm to compute one of DBSCAN's initial parameters, in addition to detecting clusters with heterogeneous density. This enables the proposed method to be more efficient and accurate.

Let $X = \{x_1, \dots, x_n\}$ be the dataset to be partitioned, where $x_i \in \mathbb{R}^d$, $\varepsilon > 0$ is the neighborhood radius and $MinPts$ is the number of points required for a region to be considered dense. The neighborhood of a point is defined in expression 9.

$$N_\varepsilon(x_i) = \{x_j \in X | d(x_i, x_j) \leq \varepsilon\} \quad (9)$$

where $d(x_i, x_j)$ is the Euclidean distance between the points x_i y x_j . A point may be considered a core point or a border point; otherwise, it is classified as noise. To determine this, expressions 10 and 11 are evaluated for each point.

$$|N_\varepsilon(x_i)| \geq MinPts \quad (10)$$

$$|N_\varepsilon(x_i)| < MinPts \text{ y } \exists x_j \text{ núcleo } x_i \in N_\varepsilon(x_j) \quad (11)$$

D. Model-Based Algorithms

Model-based clustering assumes that data are generated from an underlying probabilistic model, typically formulated as a finite mixture of parametric distributions. Each cluster corresponds to one component of the mixture, which enables flexible representation of complex and overlapping structures. These methods provide probabilistic membership assignments and are widely applied in bioinformatics, pattern recognition, and statistical learning due to their ability to incorporate uncertainty and prior knowledge [20], [21]. Given a dataset $X = \{x_1, \dots, x_n\}$ and K mixture components, the model is defined as:

$$(12) \quad p(x_i | \Theta) = \sum_{k=1}^K \pi_k f(x_i | \theta_k),$$

where π_k represents the mixing coefficients such that is the probability density of component k , and Θ denotes all model parameters. Typically using the Expectation–Maximization (EM) algorithm, which iteratively updates posterior responsibilities and parameter estimates until convergence [21].

E. Grid-Based Algorithms

Grid-based clustering algorithms partition the data space into a finite number of non-overlapping cells or grid units and perform clustering at the grid level rather than directly on individual data points. This approach significantly reduces computational complexity, as the clustering process depends on the number of grid cells instead of the number of instances. These methods are highly efficient, scalable, and robust to noise, making them suitable for large datasets or high-dimensional applications such as spatial data mining, image analysis, and multidimensional pattern detection [22], [23].

Given a dataset $X = \{x_1, \dots, x_n\}$, the data space is divided into a grid of cells, and density measures are computed for each cell to determine whether it belongs to a cluster. Grid-based clustering algorithms generally follow four steps: partitioning the space, calculating cell densities, identifying dense regions, and forming clusters by connecting adjacent dense cells.

III. COMPARATION AMONG THE MAIN FAMILIES OF CLUSTERING ALGORITHMS

Clustering algorithms can be classified according to multiple taxonomies, given the wide diversity of mathematical principles, structural assumptions, and optimization mechanisms that underpin these algorithms [24]. In general, the algorithms can be grouped into families that share common methodological foundations and similar strategies for clustering datasets. Table 1 presents the families of clustering algorithms widely used by the scientific community.

A. Partitional Algorithms

Partitional clustering algorithms are characterized by generating a single partition of data into K groups. Their operation is based on the optimization of objective functions that seek to minimize dissimilarity and maximize separation among groups [25]. These objective functions may be associated with variance, Euclidean distance, or fuzzy membership measures. Mathematically, they operate under assumptions of convexity, isotropy, and density homogeneity, which implies that the expected groups typically exhibit spherical or simple ellipsoidal geometry [26]. Most of these techniques rely on iterative processes of reassignment and updating, enabling convergence to stable solutions due to their initialization-dependent nature [27].

This family has been widely used in industrial and scientific applications owing to its efficiency, simplicity, and interpretability [28], particularly in contexts with well-separated structures or low dimensionality.

B. Hierarchy-Based Algorithms

Algorithms belonging to this class establish multilevel relationships among instances through a structure known as a dendrogram, which describes the relative similarity between elements in the dataset. Unlike partition-based clustering algorithms, they do not require specifying the number of groups and allow the analysis of clustering structure at multiple scales [16]. Their operation is based on two strategies: agglomerative, which begins with each data point as an individual group; and divisive, which starts with a global group and successively divides it. The key operational criterion is the linkage function, whose selection directly affects the shape of the dendrogram and the final structure of the groups. Mathematically, hierarchical algorithms work with distance matrices, providing an appropriate representation for clustering, though computationally costly [29]. These methods are useful in exploratory analysis, bioinformatics, and social sciences, where interpretability is a critical factor [17].

C. Density-Based Algorithms

Clustering algorithms can be classified according to data density; that is, groups are defined as regions in the space where data concentration is higher compared to surrounding areas [30]. This formulation derives from notions of topology, graph theory, and density connectivity. Generally, these methods identify dense connected components within a geometric graph constructed using metric distances. They are suitable for non-convex structures and datasets containing noise. These methods assume that a cluster is a densely connected region in which each point has a minimum number of neighbors within a given radius; however, high sensitivity to global parameters may lead to unstable results. Recent studies report their effectiveness in spatial and geographic analysis for detecting dense regions in GPS data and urban structures [18], as well as in contemporary anomaly-detection tasks applied to network traffic, cybersecurity, and sensor monitoring in IoT environments [31], [32].

D. Grid-Based Algorithms

This type of clustering algorithm quantifies the data space into a finite number of cells or grids, within which local

statistics are computed to represent the regional distribution of the data. This approach reduces computational complexity, as the dataset is scanned only once to construct the grid, while subsequent clustering occurs in a transformed space whose size depends on the selected resolution and the total number of instances. Notable advantages arise in terms of efficiency, scalability, and noise resistance, making these methods suitable for large or high-dimensional databases. The quality of the clustering depends on the size of the cells into which the data space is divided. The literature reports its application in spatial data mining, geographic analysis, and the processing of information structured in subspaces [33], [34].

E. Model-Based Algorithms

This type of algorithm formulates the clustering problem as a parameter estimation task, where each cluster is interpreted as a component of a mixture distribution; the dataset is modelled as a weighted combination of parametric distributions, allowing the underlying statistical structure of the groups present in the data to be represented [35]. A distinguishing characteristic of this family is the geometric flexibility derived from the parameterization of covariance matrices, which may assume spherical, diagonal, elliptical, or freely oriented forms [36]. Recent research highlights the growth of Bayesian models, particularly Bayesian Gaussian Mixtures, which are used in scenarios with limited labelled data, complex structures, and the need to incorporate prior knowledge [37].

The broad and growing variety of clustering algorithms characterized by diverse theoretical foundations, statistical assumptions, and operational mechanisms has turned the selection of the most appropriate algorithm for a specific problem into a methodological challenge. In this context, the construction of a formal taxonomy of clustering algorithms emerges as a fundamental tool for structuring existing knowledge. In this way, classifying algorithms into coherent conceptual families can establish a reference framework that facilitates their systematic comparison and justified application in particular scenarios [38]. Table 1 presents the algorithms that make up the partitional family, including methods widely used for their efficiency and suitability for exploratory analysis in large datasets. Table 2 includes the algorithms belonging to the hierarchical family, recognized for constructing nested relationships among data through divisive or agglomerative processes.

Table 1. Taxonomy of partition-based clustering algorithms.

Algorithm	Metrics Used	Advantages	Disadvantages	Usage Scenarios	Ideal Dataset Characteristics
K-means	Euclidean, Cosine	Fast and efficient; easy to implement.	Sensitive to outliers; requires k ; convex clusters.	Numerical data, rapid segmentation.	Numerical variables; spherical clusters; no outliers.

“Clustering algorithms: taxonomies, comparative studies, and emerging approaches based on meta-features of datasets”

K-medoids	Euclidean, Manhattan, Hamming	Robust to outliers; supports multiple metrics.	More computationally expensive.	Mixed data.	Mixed variables; presence of outliers.
PAM	Euclidean, Manhattan, Hamming	Highly robust; high quality for small datasets.	High computational cost.	Qualitative or specialized analysis.	High precision; small and noisy datasets.
CLARA	Euclidean, Manhattan, Hamming	More scalable than PAM.	Depends on sampling.	Large-scale data mining.	Representative samples; homogeneous distribution.
CLARANS	Euclidean, Manhattan, Hamming	Balance between speed and accuracy.	Increased randomness.	Semi-automated clustering.	Complex data; interactive exploration.
MiniBatch K-means	Euclidean	Very fast; scalable; low memory usage.	Lower accuracy.	Streaming, big data, online recommendations.	Large volumes; stable streaming data.
Algorithm	Metrics Used	Advantages.	Disadvantages.	Usage Scenarios.	Ideal Dataset Characteristics.

Table 2. Taxonomy of hierarchy-based clustering algorithms.

Algorithm	Metrics Used	Advantages	Disadvantages	Usage Scenarios	Ideal Dataset Characteristics
Agglomerative Hierarchical	Linkage distance	Does not require k initially; dendrogram visualization.	High computational cost; sensitive to outliers.	Structural exploration; biological or linguistic studies.	Numerical data with clear hierarchical structure.
Divisive Hierarchical	Linkage distance	Explores large hierarchical structures.	Less supported in libraries; results sensitive to early decisions.	Community analysis or hierarchical classification.	Nested, non-spherical structures.
Ward’s Method	Intra-cluster variance; Euclidean distance	Reduces variance; produces compact and balanced clusters.	Assumes spherical shape; not suitable for elongated clusters.	Clustering in psychometrics and marketing studies.	Homogeneous data; expected compact clusters.

Table 3 groups the algorithms corresponding to the density-based family, characterized by identifying regions of high data concentration. Table 4 covers the algorithms that form part of the model-based family, oriented toward representing data through statistical distributions.

Table 5 compiles the algorithms that constitute the grid-based family, which partition the space into discrete cells to operate on local aggregations.

Table 3. Taxonomy of density-based clustering algorithms.

Algorithm	Metrics Used	Advantages	Disadvantages	Usage Scenarios	Ideal Dataset Characteristics
DBSCAN	Euclidean distance (customizable)	Identifies arbitrarily shaped clusters; detects noise.	Sensitive to parameters; difficulty in high-dimensional spaces.	Spatial databases, anomaly detection.	Variable density; non-spherical clusters; presence of noise.
OPTICS	Euclidean distance or user-defined metric	Does not require specifying the number of clusters; handles variable densities.	High computational complexity; requires interpretation of the Reachability plot.	Data with density variability; detailed exploration.	Overlapping clusters; heterogeneous densities.

HDBSCAN	Euclidean distances; density-based metrics	Improved detection of complex structures and noise; natural hierarchy.	Requires fine-tuning; results may be less intuitive.	Data science, bioinformatics, geospatial analysis.	Large hierarchical structure.	volumes; density
---------	--	--	--	--	-------------------------------	------------------

Table 4. Taxonomy of model-based clustering algorithms.

Algorithm	Metrics Used	Advantages	Disadvantages	Usage Scenarios	Ideal Dataset Characteristics
Gaussian Mixture Models (GMM)	Log-likelihood probability, Mahala Nobis distance	Allows elliptical cluster shapes; probabilistic.	Sensitive to initialization; requires predefined k	Probabilistic classification, NLP, image segmentation.	Continuous data with overlapping clusters.
Hidden Markov Models (HMM)	Transition and emission probabilities	Models temporal sequences; detects patterns.	High training complexity; requires ordered sequences.	Speech recognition, time series analysis.	Temporal or sequential data.
Bayesian Gaussian Mixture	Posterior distribution, Bayesian criteria	Prevents overfitting; incorporates uncertainty.	Higher computational cost; statistical complexity.	Problems requiring prior knowledge integration.	Complex structures and limited labeled information.

Table 5. Taxonomy of grid-based clustering algorithms.

Algorithm	Metrics Used	Advantages	Disadvantages	Usage Scenarios	Ideal Dataset Characteristics
STING (Statistical Information Grid)	Cell density; aggregated statistics	Fast and scalable; low computational cost	Fixed resolution; loses precision in complex structures	Spatial data mining; range queries	Geographic or spatially structured data
CLIQUE (Clustering in Quest)	Density in subspaces; number of points per cell	Identifies clusters in high-dimensional subspaces	Sensitive to cell size; not always intuitive	Multidimensional databases; bioinformatics	High-dimensional data with hidden clusters
Wave Cluster	Frequency energy; implicit spatial distance	Captures arbitrary shapes; robust to noise	Complex to implement; requires specialized parameters	Image processing, signal analysis, spatial pattern detection	Spatial data with noise or fractal structure

IV. REFERENCE DATASETS

The comparison and analysis of clustering algorithms require standardized datasets that allow their behavior to be studied under controlled conditions. In the literature, repositories such as the UCI Machine Learning Repository [39], OpenML [40], and KEEL [41] are relevant due to the variety of datasets they provide.

A. UCI Machine Learning Repository

The UCI Machine Learning Repository is one of the most influential and widely used repositories by the scientific community since its creation in 1987 [39]. Its original purpose was to provide an accessible collection of datasets

for researchers interested in the empirical analysis of machine learning and statistical algorithms. Over time, it has become a reference resource for the development, testing, and comparison of both supervised and unsupervised techniques, including clustering algorithms.

The UCI repository provides the scientific community with a wide variety of datasets organized according to different criteria, such as task type, application domain, and attribute characteristics. This structure makes it easier for researchers to select appropriate datasets for evaluating and comparing clustering algorithms based on the specific needs of their experiments.

Among the most representative datasets are those listed in Table 6, which have been widely used in the literature due to their public availability, the quality of their documentation, their consistent presence in comparative studies, and their historical relevance as benchmarks for evaluating clustering algorithms.

Table 6: Most frequently used datasets.

<i>Datasets</i>	<i>Instances</i>	<i>Attri- butes</i>	<i>Description</i>
Iris	150	4	Flower morphology; classic dataset for clustering validation.
Wine	178	13	Chemical analysis of wines; widely used in partitional clustering.
Breast Cancer Wisconsin	569	30	Cancer diagnosis; used in clustering studies and external metrics.
Seeds	210	7	Geometric characteristics of grains; well-defined groups.
MNIST	70,000	784	Image dataset; widely used in clustering, dimensionality reduction, and advanced methods.

B. OpenML

The OpenML data repository [40] is a collaborative platform designed to facilitate reproducible experimentation in machine learning. Unlike traditional repositories, OpenML not only stores datasets but also integrates metadata, structured descriptions, predefined tasks, workflows, evaluations, and experimental results generated by the scientific community. This architecture, oriented toward traceability and standardization, has made OpenML a key resource for research requiring consistent databases and systematic experimentation in the field of clustering.

One of the distinctive features of OpenML is its ability to provide standardized versions of datasets, which allows clustering techniques to be compared under controlled and replicable conditions. The platform maintains a detailed record of each dataset’s properties, including the number of instances, attributes, variable types, complexity measures, basic statistics, and automatically computed meta-features. This approach is particularly relevant for research involving meta-learning, algorithm recommendation models, and analyses based on meta-characteristics.

B. KEEL

The KEEL repository [41] offers an extensive collection of databases specifically oriented toward machine learning,

with a particular emphasis on unsupervised problems, evolutionary methods, and hybrid techniques. Unlike traditional repositories, KEEL includes preprocessed, normalized, and partitioned versions of each dataset, which facilitates the execution of rigorous and fully reproducible comparative analyses. Among its most notable contributions, KEEL provides:

- Datasets designed for clustering, feature synthesis, and dimensionality reduction.
- Standardized versions of each dataset under different preprocessing levels (normalization, discretization, noise removal), allowing the evaluation of algorithm behavior under controlled scenarios.
- Detailed metadata, including statistical information, number of classes (when applicable), complexity levels, presence of noise, and attribute structure.
- Special collections, such as hard datasets used for benchmarking evolutionary algorithms, density-based methods, or hierarchical clustering techniques.

The UCI, OpenML, and KEEL repositories constitute the main sources for evaluating clustering algorithms, each providing complementary contributions. UCI offers classic and well-documented datasets; OpenML provides standardized versions with complete metadata; and KEEL integrates preprocessed datasets for rigorous comparative analyses. This diversity facilitates reproducible experiments and strengthens the systematic study of algorithmic performance, while also supporting emerging approaches based on meta-characteristics and meta-learning aimed at intelligent algorithm selection.

V. CLUSTERING APPLICATIONS

Clustering algorithms have a wide range of applications across various domains, where pattern identification, element segmentation, and structural data organization are essential for understanding complex phenomena. In the health domain, clustering has been used to identify clinical profiles, segment patients according to medical characteristics, and characterize disease patterns, supporting diagnosis and therapeutic personalization [42], [43].

In the field of education, clustering techniques enable the creation of learning profiles, the detection of interaction patterns in virtual environments, the classification of academic trajectories, and the analysis of student behaviors, contributing to the development of recommendation systems and learning analytics [44]. In finance and economics, clustering is used to segment customers, identify risk profiles, detect anomalous behaviors, and analyze patterns in economic transactions, strengthening data-driven decision-making and financial risk management [45].

In the domain of social network analysis, clustering has proven useful for community detection, grouping users with similar behaviors, and identifying emerging topics, enabling the understanding of social dynamics and global information

trends [46]. Finally, in the field of security, these techniques play a key role in detecting anomalous patterns, identifying suspicious activities, analyzing network traffic, and strengthening behavior-based intrusion detection systems [47].

VI. CONCLUSIONS AND FUTURE WORK

The state of the art in clustering algorithms reveals a broad and diverse field, structured into five main families: partitional, hierarchical, density-based, model-based, and grid-based. Each family presents its own characteristics and accepted data types; this methodological diversity confirms the absence of a universally superior algorithm and underscores the importance of considering dataset characteristics in the selection process.

The UCI Machine Learning, OpenML, and KEEL repositories, together with the datasets widely used by the scientific community such as: Wine, Breast Cancer Wisconsin, and MNIST, constitute the most frequently used experimental basis in the literature. Their public availability, clear documentation, and ease of use provide methodological consistency, reproducibility, and comparability across studies.

Clustering applications in domains such as health, finance, economics, social networks, security, among others, demonstrate its presence across multiple scientific and technological areas. The ability to group information and support data-driven decision-making reinforces its importance in Data Science.

The combination of established taxonomies, standardized datasets, and a wide range of applications provides a solid foundation for future developments and continuous improvement of clustering techniques within Data Science.

In this context, future work is oriented toward the development of more precise taxonomies, large-scale comparative studies, and approaches based on dataset meta-characteristics that enable the selection of clustering algorithms. Likewise, growth is anticipated in hybrid methods and advanced probabilistic approaches, along with recommendation systems that automate the choice of the most suitable algorithm according to the structure and complexity of the dataset. These lines of research represent a promising trajectory for the use of clustering techniques and their integration into increasingly complex scenarios within Data Science.

ACKNOWLEDGMENT

The Student Yesenia Santana Cardoso Acknowledges his scholarship (grantee No. 4007298) To the Secretaría de Ciencia, Humanidades, Tecnología e Innovación. This research was partially funded by the National Technology of Mexico with the project: 21639.25-P.

REFERENCES

1. Liu, Q., Liu, J., Li, M., & Zhou, Y. (2021). Approximation algorithms for fuzzy C-means problem based on seeding method. *Theoretical Computer Science*, 885, 146-158.
2. Aalst, W. (2016). *Process Mining. En Data Science in Action* (págs. 3-23). Springer.
3. Provost, F., & Fawcett, T. (2013). Data Science and Its Relationship to Big Data and Data-Driven Decision Making. *Big data*, 1(1), 51–59.
4. Pitafi, S., Anwa, T., & Sharif, Z. (2023). A Taxonomy of Machine Learning Clustering Algorithms, Challenges, and Future Realms. *Applied Sciences*, 13, 3529-3529.
5. Dinh, T., Wong, H., Fournier-Viger, P., Lisik, D., Ha, M.-Q., Hieu Chi, D., & Van-Nam, H. (2025). Categorical data clustering: 25 years beyond K-modes. *Expert Systems with Applications*, 272, 2-49.
6. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281-297.
7. Pérez, J., Montes, P., Mexicano Santoyo, A., & Rodriguez-Jorge, R. (2017). Acceleration of the K-means algorithm by removing stable items. *International Journal of Space-Based and Situated Computing*, 7, 72-81.
8. Pérez Ortega, J., Almanza-Ortega, N., & Romero, D. (2018). Balancing effort and benefit of K-Means clustering algorithms in Big Data realms. *PLOS One*, 13(9), 1-19.
9. Pérez-Ortega, J., Hidalgo-Reyes, M., Castro-Sánchez, N., Pazos-Rangel, R., Díaz-Parra, O., Olivares-Peregrino, V., & Almanza-Ortega, N. (2018). An Efficient Heuristic Applied to the K-means Algorithm for the Clustering of Large Highly-Grouped Instances. *Computing and Systems*, 2.
10. Almanza-Ortega, N., Pérez-Ortega, J., Zavala-Díaz, J., & Solís-Romero, J. (2022). Comparative Analysis of K-Means Variants Implemented in R. *Computing and Systems*, 26(1).
11. Bezdek, J., Ehrlich, R., & Full, W. (1984). FCM: The Fuzzy c-Means clustering algorithm. *Computers & Geosciences*, 10(2), 191-203.
12. Pérez Ortega, J., Roblero Aguilar, S., Almanza-Ortega, N., & Frausto, J. (2022). Hybrid Fuzzy C-Means Clustering Algorithm Oriented to Big Data Realms. *Axioms*, 11, 377.
13. Wu, Z., Yao, J., & Chen, G. (2019). The Stock Classification Based on Entropy Weight Method

- and Improved Fuzzy C-means Algorithm. 4th International Conference, 130-134.
14. Siringoringo, R., & Jamaluddin, J. (2019). Initializing the Fuzzy C-Means Cluster Center With Particle Swarm Optimization for Sentiment Clustering. *Journal of Physics Conference Series*, 1361.
15. Liu, X.-y., Fan, J., & Chen, Z.-w. (2020). Improved fuzzy C-means algorithm based on density peak. *International Journal of Machine Learning and Cybernetics*, 11, 545-552.
16. Hierarchical clustering schemes. (1967). En Publishing model (Vol. 32, págs. 241–254). Springer.
17. Mogharabi, A., Taghizadeh, A., & Kalantari, K. (2024). HACCR: Hierarchical Agglomerative Clustering-Based Connectivity Restoration Approach in Mobile Wireless Sensor and Robot Networks. *Wireless Personal Communications*, 138, 1773-1798.
18. Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226-231.
19. An Adaptative Eps Parameter of DBSCAN Algorithm for Identifying Clusters with Heterogeneous Density. (2024). *Computing and systems*, 28(2), 465-472.
20. Basford, G. J. (1988). Mixture Models: Inference and Applications to Clustering. *Journal of the Royal Statistical Society Series C Applied Statistics*, 38.
21. Raftery, C. F. (2002). Model-based clustering, discriminant analysis, and density estimation. 97(458), 611–631.
22. W. Wang, J. Y. (1997). STING: A statistical information grid approach to spatial data mining. in *Proc. 23rd Int. Conf. Very Large Data Bases (VLDB)*, 186–195.
23. G. Sheikholeslami, S. C. (1998). WaveCluster: A multi-resolution clustering approach for very large spatial databases. *Proc. 24th Int. Conf. Very Large Data Bases (VLDB)*, 428–439.
24. Zhou, S., Xu, H., Zheng, Z., Chen, J., li, Z., Bu, J., Ester, M. (2024). A Comprehensive Survey on Deep Clustering: Taxonomy, Challenges, and Future Directions. *ACM Computing Surveys*, 1-38.
25. Partitional Clustering. (2011). En X. Jin , & J. Han , *Encyclopedia of Machine Learning* (pág. 799). Springer, Boston, MA.
26. Chicco, D. (2024). Comprehensive analysis of clustering algorithms: exploring. *PeerJ Computer Science*, 45.
27. Wang, S., Sun, Y., & Bao, Z. (2020). On the efficiency of K-means clustering: evaluation, optimization, and algorithm selection. *Proceedings of the VLDB Endowment*, 163-175.
28. Jain, A. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31, 651-666.
29. Murtagh, F., & Contreras, P. (2011). Algorithms for hierarchical clustering: an overview. *WIREs Data Mining and Knowledge Discovery*, 2(1), 86-97.
30. Sander, J. E. (1998). Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. En *Data Mining and Knowledge Discovery* (Vol. 2, págs. 169–194). Springer.
31. Kut, M. B. (2007). ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, 60(1), 208–221.
32. Lohiya, A. T. (2019). Survey on anomaly detection in network traffic using machine learning. *Procedia Computer Science*, 165, 636–643.
33. S. Shekhar, Z. J. (2019). Spatiotemporal data mining: A computational perspective. *IEEE Transactions on Knowledge and Data Engineering*, 31(9), 1763–1778.
34. M. Mallik, O. D. (2020). A grid-based clustering approach for high-dimensional data analysis. *IEEE Access*, 8, 74268–74282.
35. Wainwright, R. Y. (2020). Structured Bayesian Gaussian mixture models for clustering high-dimensional data. *IEEE Transactions on Information Theory*, 66(4), 2566–2598.
36. McNicholas, M. B. (2021). Model-based clustering with sparse and irregular data: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10), 3534–3551.
37. Murphy, S. M. (2022). Model-based clustering overview. *Annual Review of Statistics and Its Application*, 9, 195–223.
38. C. Ruiz, M. S. (2022). Survey on clustering: Taxonomy, methods, and empirical analysis. *IEEE Transactions on Knowledge and Data Engineering*, 34(7), 2763–2787.
39. Aha, D. (s.f.). UC Irvine Machine Learning Repository. <https://archive.ics.uci.edu/>
40. Vanschoren, J. (s.f.). OpenML. (OpenML) <https://openml.org/>
41. Alcala-Fdez., J. (s.f.). KEEL-dataset repository. KEEL-dataset repository: <https://sci2s.ugr.es/keel/datasets.php>
42. Loftus, T. J.-B. (2022). Phenotype clustering in health care: A narrative review for clinicians. *Frontiers in artificial intelligence*, 5.
43. Jentzer, J., Reddy, Y., Soussi, S., Crespo-Diaz, R., Patel, P., Lawler, P., . . . M. Dunlay, S. (2024). Unsupervised machine learning to identify

- subphenotypes among cardiac intensive care unit patients with heart failure. *ESC Heart Failure*, 11(6), 4242-4256.
44. Franke, S., Trommer, M., Hellbach, S., & Teich, T. (2024). Clustering of Learners' Data – A Case Study on Learning Analytics. *IFAC-PapersOnLine*, 58(26), 118-123.
 45. Pratama, S. (2024). User Profiling Based on Financial Transaction Patterns: A Clustering Approach for User Segmentation. *International Journal for Applied Information Management*, 4, 217-228.
 46. Nooribakhsh, M., Fernández-Diego, M., González-Ladrón-de-Guevara, F., & Mollamotalebi, M. (2024). Community detection in social networks using machine learning: a systematic mapping study. *Knowledge and Information Systems*, 66, 7205-7259.
 47. N. Sushmakar, N. O. (2022). An Unsupervised Based Enhanced Anomaly Detection Model Using Features Importance. *2nd International Conference on Intelligent Technologies (CONIT)*, 1-7.