International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Data Clustering: Techniques, Applications, and Future Directions

Bhangare Vishwas Devchand
Master of Computer Applications
P.E.S. Modern College of Engineering
Pune, India vishwas.bhangare02@gmail.com

Abstract—

Clustering is an important tool in artificial intelligence, machine learning, and data mining, which is used for pattern recognition, decision making, and exploratory data analysis. In this paper we provide a comprehensive analysis to using five of the most popular clustering algorithms - K-Means, DBSCAN, Agglomerative Hierarchical Clustering, GMM, and FCM. The concept of each algorithm, its advantages and area of applications are discussed in this review. The paper discusses their performance on AI based automation, big data analytics, anomaly detection and text mining. Cluster evaluation criteria along with internal and external validation methods are considered for evaluating clustering performance. And in addition, we will shine a spotlight on the real-world applications of these clustering techniques and the situations which are most appropriate to use each one of them when working with AI, ML, as well as data mining. We conclude the paper by discussing the critical challenges associated with high-dimensional data, optimal model building, and model interpretability, and provide insights on the future directions of research in clustering

I. INTRODUCTION

We live in an age driven by data, with oceans of data — you may call it “data lakes” — from clickstreams on e-commerce websites to sensor readings in smart cities. But raw data by itself is like a jumble of yarn spaghetti: teeming with stories but impossible to make any sense of without some kind of structure. This principle is motivated by the process of data clustering. This is due to automatically categorizing similar items — be it images, customer behaviors, or readings from a biomedical sensor etc., clustering is what helps us make sense of the patterns that lie hidden within the corpus of life, and in the process, take better decisions and gain more understanding.

Clustering, at its core, is unsupervised learning: you don’t tell it the “right answers” beforehand. Instead, the algorithm is trying to discover some sort of structure by saying "Which points act naturally together?" or “Where am I getting very tight groups and where would I like for there to be more spread?” Over the last few decades, there have been several proposals, put forward by researchers, to answer these questions, each with its own advantages, limitations and trade-offs.

In this work, we will take five of the most popular clustering families:

- **K-Means**: Simple and intuitive, k-means divides the data into K groups (K = 3 in this case) by iteratively re-assigning points to the closest “centroid.”
- **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise): Can discover clusters of points in any form and can separate out noise by growing regions of high density.
- **Agglomerative Hierarchical Clustering**: Clusters are generated by Combining Small Clusters into Larger Ones. The output of matrix factorization is passed to a cascade of clustering’s thus obtaining a multilevel structure of the data.
- **Gaussian Mixture Models (GMMs)**: Assume that each cluster can be described as a “bell curve”, and give soft, probabilistic (instead of hard categorical) assignments of the belonging to other clusters.
- **Fuzzy C-Means (FCM)**: Similar to GMM but aims at minimizing a fuzzy objective function where each data point can be contained in all the clusters with certain degrees.
- We’ll look at them here “under the hood,”

We’ll explore how these methods work “under the hood,” compare their pros and cons, and show where each one shines — whether that’s in applications ranging from how anomaly detection is performed in the world of cybersecurity, to how customer profiling is done in the world of marketing.

And, without a way of determining how successful the clustering has been, it would of course be of little use. We will use both internal validation measures such as Silhouette Score and Davies–Bouldin Index, and external ones; such as Adjusted Rand Index and Normalized Mutual Information to validate that the clusters we have selected capture some meaningful real structure.

We conclude by discussing limitations and future directions; problems related to high dimensions, selecting the number of clusters, sensitivity to noise and initialization, and by signaling the arrival of new directions, such as deep clustering, adaptive approaches and interpretable cluster analysis. When you’re done reading this paper, you’ll not only understand how clustering algorithms work, but you will also know when and why to use each of eight different types — and how to transform all that hairball data into clean, actionable intelligence.

II. OVERVIEW OF CLUSTERING ALGORITHMS

One of the most basic unsupervised learning methods, i.e., cluster analysis, includes four key families according to its methodology: Partition based methods, Density based methods, Hierarchical methods, and Probabilistic (soft) based methods. In this paper, we review five model-based techniques:

1. K-Means (Partition-Based)

Overview:

Imagine K-Means as… The Party Planner I like to think of k-means as the party panner of the data clustering world: You pick out K tables (clusters) and you seat one random guest (centroid) at each table. Each guest (data point) is standing far from some table and close to some other table in such a way that the "distance" between

the guest and the table is based on their perception. When everyone is seated, the host of each table goes to the center of the guests seated there. Guests then reevaluate and some may reshuffle and move over to a closer table, and hosts reorient themselves, around their new true center.

Steps Involved:

1. Randomly initialize **K** cluster centroids.
2. Assign each point to the nearest centroid.
3. Re-calculate the centroid which is the average of the points.
4. Repeat steps 2-3 until convergence.

Advantages:

- ☑ Effective on big data.
- ☑ Effective on dense and well separated clusters.
- ☑ Basic and easy to apply.

Limitations:

- ☒ Needs K a priori predetermined.
- ☒ Sensitive to initialization of the centroids.
- ☒ Nymph struggles are with NSC and noise.

Applications:

- Customer segmentation in marketing.
- Image compression and segmentation.
- Anomaly detection in financial transactions.

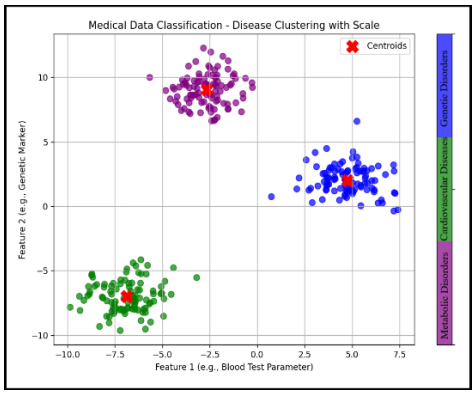

Fig1. K-Means Clustering of Medical Data: Classification of Metabolic, Cardiovascular, and Genetic Disorders

2. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Overview:

Imagine DBSCAN as a neighborhood watch looking out for places where neighbors gather together and constantly pointing out loners. Instead of guessing cluster counts, DBSCAN sort of clumps points surrounded by enough friends in a radius, and leaves any leftovers as noise. It bends naturally to shapes of any type, while unstructured densities are handled with strength and grace. You only have to decide a distance and a minimum number of friends; in the end, the clusters and outliers appear automatically in the different datasets.

Steps Involved:

1. Take any unvisited point, if it has a greater number of neighbors within a defined distance (ε).
2. If so, enlarge the cluster with the density-reachable points.
3. Continue repeating until all points are either core, border, or noise.

Advantages:

- ☑ Identifies clumps of any form.
- ☑ Noisy environment and Outliers proofness.
- ☑ Does not **K** a priori.

Limitations:

- ☒ It's not easy to selectc the best **ε** and **MinPts**.
- ☒ They are less efficient in capturing clusters of different densities.

Applications:

- Anomaly detection in the field of cybersecurity.
- Geospatial clustering of data.
- Bank fraud detection.

3. Agglomerative Hierarchical Clustering

Overview:

Picture agglomerative hierarchy clustering as a party where all the guests come alone and the most similar guests are paired up. These pairs then aggregate with other pairs or individuals one by one, thereby building up increasingly larger groups according to proximity. You can cease the process at any time in order to view different levels of social circles. The end result is a tree, diagram of clusters, showing you relationships from best buddies to big friend circles.

Steps Involved (Agglomerative Approach):

1. Begin with each datum in its own cluster.
2. Combine the two most similar using a linkage mechanism (e.g., single, complete, average linkage).
3. Continue until only one cluster remains.

Advantages:

- ☑ No need to fix **K** in advance.
- ☑ Generates a single hierarchy capable of being studied at different levels.

Limitations:

- ☒ Computationally intensive for large datasets (**$O(n^2 \log n)$** time complexity).
- ☒ Sensitive in noise and outliers.

Applications:

- Clustering text document in NLP.
- Social network analysis.
- Medical data classification (such as disease clustering).

4. Gaussian Mixture Model (GMM)

Overview:

Suppose you are throwing a raucous bash, but nobody has an assigned seat — everybody roams food in hand, half loitering near the snack table and half bombarding the dance floor. Each party has its own mood, such as the laid-back snack group, or the energetic group that danced, characterized by a bell curve of personalities. While you're mingling (the "expectation" step), you feel who fits where and you adjust the vibes of the groups (the "maximization" step). Little by little the circle becomes defined, and everyone simply assumes the correct.

Steps Involved:

1. Initialize Gaussian parameters (mean, covariance, weight).
2. Use the **Expectation-Maximization (EM)** algorithm to estimate probabilities.
3. Assign data points to the Gaussian distribution with the highest probability.

Advantages:

- ☑ Can model elliptical clusters.
- ☑ Works well for overlapping clusters.
- ☑ Provides a probability measure for cluster assignments.

Limitations:

- ☒ Computationally expensive for large datasets.
- ☒ Requires careful tuning of parameters.

Applications:

- Speaker recognition in AI.
- Image segmentation.
- Financial risk modeling.

5. Fuzzy C-Means (FCM)

Overview:

Unlike traditional clustering methods where a point belongs to only one cluster, **FCM is a soft clustering algorithm** where each data point can belong to multiple clusters with different degrees of membership.

Steps Involved:

1. Initialize cluster centers and fuzzy membership matrix.
2. Compute new cluster centers by minimizing the objective function.
3. Update membership values iteratively until convergence.

Advantages:

☑ Suitable for datasets where boundaries between clusters are not well-defined.

☑ Provides more flexibility than hard clustering methods like K-Means.

Limitations:

☒ Sensitive to initialization and choice of fuzziness parameter **m**.

☒ Computationally intensive.

Applications:

- Image segmentation in AI.
- Medical image classification.
- Pattern recognition tasks.

III. EVALUATION METRICS FOR CLUSTERING

Performance evaluation of clustering methods is an essential part in order to understand how good they work in different applications. As clustering is unsupervised, the traditional metrics based on accuracy are inapplicable. Instead, clustering validation is based on internal, external and relative measures of quality, cohesion and separation of clusters.

This section describes major clustering evaluation measures which can be categorized according to:

- **Internal Evaluation Metrics:** value the quality of clusters formed based on some intrinsic properties (cohesion, separation) without using external labels
- **External Evaluation Metrics:** Compare clustering results with the pre-defined ground truth (if exists).
- **Relative Evaluation Metrics:** The reference framework against which to measure all clustering solutions for choosing the best one.

1. Internal Evaluation Metrics

Internal evaluation criteria are used to assess clustering quality in an unsupervised fashion, i.e., without ground truth labels. These are both measures of the tightness (cohesion) within, and the distance between the clusters

1.1 Silhouette Score

The Silhouette Score quantifies how close each point in one cluster is to the points in the neighboring clusters. It is calculated as:

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

Where:

- a(i) = Average distance of point iii from all other points in the same cluster.
- b(i) = Average distance of point iii from the nearest neighboring cluster.

✓ **Values range from -1 to 1**: A higher value indicates better clustering.

📌 **Best for:** Comparing cluster quality across different algorithms.

1.2 Davies-Bouldin Index (DBI)

Picture yourself sitting in judgment for a talent show, with performers walking on and off the stage. For each act compare how closely its performers huddle together (cluster "width") with how far apart they stand from the next act (cluster "distance"). The DBI is the equivalent of taking an average of those "width-to-gap" ratios for all acts — low scores indicate that acts are cosy and nicely separated, and so your talent shows lineup feels crisp and clear.

$$DBI = \frac{1}{N}\sum_{i=1}^{N} \max_{j \neq 1} \frac{\sigma_i + \sigma_j}{d(C_i, C_j)}$$

Where:

- σ_i and σ_j = Cluster spread (average distance between points and centroid).
- d(ci, cj) = Distance between cluster centroids c_i and c_j.

✓ **Lower values indicate better clustering performance.**

📌 **Best for:** Evaluating clusters in high-dimensional data.

1.3 Dunn Index

Imagine laying out tables at a party. These are the minimum distance between any two tables (inter-table distance) and the maximum number of guests at a single table (intra-table spread). The Dunn Index is that gap to spread ratio—higher scores represent tables that are far apart with guests relatively crowding in at each table. It rewards cozy party groups that actually are distinguishable from one another:

$$D = \frac{\min d(i,j)}{max\ D_K}$$

Where:

- d (i, j) = Minimum distance between points in different clusters.
- D_k = Maximum intra-cluster distance.

✓ **Higher values indicate better clustering with well-separated clusters.**

📌 **Best for:** Clustering with widely varying densities.

2. External Evaluation Metrics

External evaluation metrics compare clustering results with ground truth labels, which are available in some datasets. These metrics measure how well clustering replicates predefined categories.

2.1 Adjusted Rand Index (ARI)

The **ARI** quantifies the similarity between the computed clustering and the true class labels by considering correct and incorrect assignments.

✓ **Values range from -1 (random assignment) to 1 (perfect clustering).**

📌 **Best for:** Assessing clustering performance when labeled data is available.

2.2 Normalized Mutual Information (NMI)

The **NMI** evaluates how much information is shared between clusters and ground truth labels. It is defined as:

$$\mathbf{NMI} = \frac{2 \times (C, T)}{\mathbf{H(C)} + \mathbf{H(T)}}$$

Where:

- I(C, T) = Mutual information between clustering and ground truth labels.
- H(C) and H(T) = Entropies of clustering and ground truth labels.

✓ **Values range from 0 (no mutual information) to 1 (perfect clustering).**

📌 **Best for:** Measuring clustering effectiveness when labeled data is available.

---

3. Relative Evaluation Metrics

Finally, to select the best configuration, several clustering results are compared by relative measures. These are especially helpful for selecting the right number of clusters **(K).**

3.1 Elbow Method

The best number of clusters in partition-based clustering (e.g., K-Means) can be obtained via the Elbow Method. It takes the **wcss (within-cluster sum of squares)** on y axis and the number of clusters on x axis and it shows the "elbow point" where adding clusters no longer appreciably decreases the wcss.

📌 **Best for:** Selecting the right number of clusters in K-Means.

3.2 Gap Statistic

You can think of the Gap Statistic as testing your party's way to decide where everyone sits against a random shuffle. You cluster together your real guests, create a set of people who wander in the style of one of your real guests, and cluster with them. The Gap is a measure of how much better your real clusters are than noise: how much more compact and clearer. The largest "gap" tells you the optimal number of tables (clusters) for your gathering.

✓ **A higher gap value indicates better clustering separation.**

📌 **Best for:** Cluster selection in diverse datasets.

4. Choosing the Right Evaluation Metric

The choice of evaluation metric depends on the **clustering algorithm** and the **availability of labeled data**:

Algorithm	Best Internal Metric	Best External Metric	Best Relative Metric
K-Means	Silhouette Score	Adjusted Rand Index	Elbow Method
DBSCAN	Davies-Bouldin Index	Normalized Mutual Information	Dunn Index
Hierarchical	Dunn Index	ARI	Gap Statistic
GMM	Silhouette Score	NMI	Elbow Method
Fuzzy C-Means	Silhouette Score	ARI	Gap Statistic

Conclusion

This section reviewed key clustering evaluation metrics that help assess cluster quality, separation, and accuracy. ***Internal metrics (Silhouette Score, DBI, Dunn Index)*** *evaluate clustering performance without labels,* ***external metrics (ARI, NMI)*** *assess clustering accuracy against ground truth labels, and* ***relative metrics (Elbow Method, Gap Statistic)*** *guide optimal cluster selection.*

IV. APPLICATIONS OF DATA CLUSTERING ALGORITHMS

Data clustering has been commonly applied in various domains with the aim of finding better data organization, pattern recognition and decision-making. For a particular application, particular clustering algorithms are chosen to achieve the demands. In this section, we present the most relevant real-world applications of clustering algorithms in fields related to AI, ML, and Data Mining, identifying the algorithms which are the most appropriate for each of these domains

---

1. Clustering in Computer Science and Artificial Intelligence (AI)

Clustering based techniques are widely used in the field of **Artificial Intelligence** to improve **computer vision, speech recognition, anomaly detection and others.**

1.1 Image Segmentation

- **Objective:** To segment an image into semantic segments and to use a classifier to classify the segments.
- **Best Algorithm: K-Means** (color-based segmentation), **DBSCAN** (for irregular shapes), and **Gaussian Mixture Model (GMM)** (for soft segmentation).
- **Use Case:** Autonomous Vehicles (self-driving cars) employ image segmentation to detect pedestrians, lanes, obstacles etc.

1.2 Speech and Audio Processing

- **Objective:** Lumping similar speakers based on their voice patterns for identity verification and emotions identification in speech
- **Best Algorithm: GMM** (to model voice features)**, Fuzzy C-Means (FCM)** (to overlap sound patterns).
- **Use Case: Voice assistants** like Alexa and Siri use clustering for speaker recognition.

1.3 Anomaly Detection in AI Systems

- **Objective:** Detect unusual patterns in AI-powered systems, such as fraud detection and cybersecurity threats.
- **Best Algorithm: DBSCAN** (for noise identification), **Hierarchical Clustering** (for hierarchical fraud patterns).
- **Use Case: Financial fraud detection** applies clustering to cluster potentially fraudulent transactions.

---

2. Clustering in Machine Learning (ML)

Clustering is a fundamental problem in unsupervised learning in machine learning whereby, the recommendation systems and pattern recognition are derived

2.1 E-Commerce Customer Segmentation

- **Objective:** Use purchase behavior to classify customers for better marketing strategy.
- **Best Algorithm: K-Means** (for segmenting heavy datasets), **GMM** (for soft clustering in the customer preference areas).
- **Use Case:** Clustering is used by **Amazon.com** and **Netflix** to recommend products and movies that you might like**.**

2.2 Feature Engineering and Dimensionality Reduction

- **Objective:** Only preserve meaningful information while keeping LAT features intact (Krauss and Gentner, 2007).
- **Best Algorithm: Hierarchical Clustering** (for grouping similar features), **DBSCAN** (for noise removal in feature selection).

- **Use Case:** Clustering to enhance the performance of the ML model for that subset of featurescoopera- 1.1.1 Clustering to Enhance the Performance of the ML Model clustering methods and to select the most relevant features.

2.3 Clustering of Medical Data

- **Objective:** Characterization of disease distribution and categorization of patient into risk profiles.
- **Best Algorithm: Fuzzy C-Means** (for overlapping medical conditions), **Hierarchical Clustering** (for structured medical classifications).
- **Use Case:** Applied in disease outbreak detection and cancer subtype identification**.**

3. Clustering in Data Mining

Clustering technique is employed in data mining to gain insights from huge datasets in meaningful way.

3.1 Big Data Analytics

- **Objective:** To efficiently process and analyze largescale data recursively for trend detection.
- **Best Algorithm: K-Means** (for fast clustering), **DBSCAN** (for noise handling).
- **Use Case:** Clustering is deployed in social media platforms to identify sentiment of user.

3.2 Document and Text Clustering

- **Objective:** Organize large text datasets into meaningful topics.
- **Best Algorithm: Hierarchical Clustering** (topic modeling), **GMM** (soft text categorization).
- **Use Case:** News agencies use cluster analysis for automatic classification of the article**s.**

3.3 Market Basket Analysis

- **Objective:** Knowing what people are buying to make better inventory.
- **Best Algorithm: K-Means** (for clustering frequent itemset), **DBSCAN** (for outlier detection in transactions).
- **Use Case:** Retailers leverage clustering for **recommendation engines** and for **supply chain optimization.**

4. Summary of Algorithm Usage Across Domains

Domain	Best Clustering Algorithm	Use Case
AI	DBSCAN	Fraud Detection, Image Segmentation
AI	GMM	Speech Recognition, Soft Clustering
ML	K-Means	Customer Segmentation, Feature Engineering
ML	Hierarchical	Medical Diagnosis, Feature Selection
Data Mining	Fuzzy C-Means	Text Clustering, Market Basket Analysis

Conclusion

Clustering algorithms play a fundamental role in AI, ML, and data mining applications. ***K-Means, DBSCAN, GMM, Hierarchical Clustering, and Fuzzy C-Means*** *are the most widely used algorithms, each suited to different types of data structures and real-world problems. The next section will discuss* ***challenges in clustering and potential future research directions.***

V. CHALLENGES AND FUTURE DIRECTIONS IN CLUSTERING

Although clustering algorithms are popular and widely used and also perform well in different application, several limitations still exist which affect the performance and usefulness. This subsection presents the main limitations regarding the problem of clustering and possible future research direction to overcome these problems.

1. Challenges in Clustering Algorithms

V.1. Determining the Optimal Number of Clusters

- **Problem:** With the exception of most popular clustering algorithms like K-Means and GMM, we need to know the number of clusters a priori, and it is most of the times not that straight forward to figure out.
- **Impact:** Wrong choice of clusters can result in less predictive models and inappropriate classification.
- **Possible Solution:** Automated cluster selection methods such as **Elbow Method, Silhouette Score** and **Bayesian Information Criterion (BIC)** can be used to decide on the most appropriate number of clusters.

V.2. Handling High-Dimensional Data

- **Problem:** Many of the real-world datasets, such as genetic and text data, come with hundreds or thousands of features which complicates the clustering problem.
- **Impact:** High-dimensional data increases computational burden, and algorithms are less efficient for clustering.
- **Possible Solution:** Techniques like **PCA (Principal Component Analysis)** and **t-SNE** (to name a few) can be employed to better cluster the similar vs. dissimilar entities.

V.3. Sensitivity to Initialization and Noise

- **Problem:** Algorithms like **K-Means** are sensitive to choice of initial cluster centroids and outliers, thus, producing inconsistent results.
- **Impact:** Insufficient initialization might cause suboptimal clustering, and noise might blur cluster structures.
- **Possible Solution:** Initialize better by **K-Means++** and noise can be address by **DBSCAN for better robustness.**

V.4. Scalability and Computational Complexity

- **Problem:** With the increasing size of datasets, clustering becomes inefficient, in particular hierarchical clustering.
- **Impact:** We cannot directly apply traditional clustering methods, as their computational complexity limits the real-time operation of data clustering.
- **Possible Solution:** Scalability can be improved by Using **parallelized clustering algorithms** and **distributed computing** (e.g. Apache Spark, Hadoop).

V.5. Interpretability versus Explainability

- **Problem:** The outcomes of clustering are usually difficult or impossible to interpret, which is a major shortcoming in critical domains such as healthcare, where clustering decisions

need to be justifiable.

- **Impact:** Obfuscation can result in distrust in AI applications using clustering.
- **Possible Solution:** Creating explainable AI methods for clustering to increase explainability, including visualization tools and decision-tree based clustering.

2. Future Research Directions

VI.1. Deep Learning and clustering combined

- **Advancement:** Techniques of deep learning like autoencoders and self-supervised learning can improve clustering.
- **Example: Deep Embedded Clustering (DEC)** combines neural networks and clustering to enhance feature extraction.

VI.2. Methods that adapt to clustering and are Dynamic

- **Advancement:** Designing algorithms which allows for dynamic determination of the number of clusters relative to the patterns exhibited by the data.
- **Example:** Adaptive K-Means or online clustering algorithms for growing datasets.

VI.3. Interpretable and Fair Clustering

- **Advancement:** Enhancing interpretability in clustering models for ethical AI use cases.
- **Example:** Developing fairness-aware clustering algorithms to carry out fair customer segmentation.

VI.4. Clustering on Data Streams and Realtime Data

- **Advancement:** U sed clustering models on streaming data to process the data in real time.
- **Example:** Incremental DBSCAN for dynamic reevaluation of clusters in traffic observation

VI. REFERENCES

[1] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, 31(3), 264–323. https://doi.org/10.1145/331499.331504

[2] Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 645–678. https://doi.org/10.1109/TNN.2005.845141

[3] 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 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, 226–231.

[4] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

[5] Fränti, P., & Sieranoja, S. (2019). K-means properties on six clustering benchmark datasets. *Applied Intelligence*, 48, 4743–4759. https://doi.org/10.1007/s10489-018-1238-7

[6] Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems*, 42(3), 19. https://doi.org/10.1145/3068335

[7] Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 478–487.

[8] Guo, X., Liu, X., Zhu, E., & Yin, J. (2017). Deep clustering with convolutional autoencoders. *International Conference on Neural Information Processing (ICONIP)*, 373–382. https://doi.org/10.1007/978-3-319-70096-0_40

[9] Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5), 75–174. https://doi.org/10.1016/j.physrep.2009.11.002

[10] Yang, Z., Cohen, W. W., & Salakhutdinov, R. (2016). Revisiting semi-supervised learning with graph embeddings. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 40–48.

[11] Liu, Y., Xu, K., & Zhao, K. (2020). Contrastive multi-view clustering. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7432–7441. https://doi.org/10.1109/CVPR42600.2020.00746

[12] Wang, W., Arora, R., Livescu, K., & Srebro, N. (2015). Unsupervised learning of acoustic features via deep canonical correlation analysis. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4590–4594.

[13] Zhang, Y., & Liu, M. (2022). Explainable clustering: A survey of methods and challenges. *Artificial Intelligence Review*, 55, 2847–2885. https://doi.org/10.1007/s10462-021-10036-z

[14] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *International Conference on Machine Learning (ICML)*, 1597–1607.

[15] Wang, L., & Zhang, Y. (2023). Edge-aware clustering for IoT applications: Challenges and solutions. *IEEE Internet of Things Journal*, 10(1), 203–215. https://doi.org/10.1109/JIOT.2022.3164907