

A comparative study of traditional, deep, and explainable clustering algorithms for behavioral segmentation of MOOC learners

Doreyed Muhammed Ahmed Al-Kerboly ^{1*}

¹College of Education for Pure Science, University Of Anbar, Anbar, Iraq

ARTICLE INFO

Received: 30/08/2025
Revised: 18/10/2025
Accepted: 28/10/2025
Available online: 08/08/2026
August Issue
[10.37652/juaps.2025.164496.1649](https://doi.org/10.37652/juaps.2025.164496.1649)

 CITE @ JUAPS

Corresponding author
Doreyed Muhammed Ahmed
doreyedm@uoanbar.edu.iq

ABSTRACT

The rapid growth of Massive Open Online Courses (MOOCs) has generated an enormous amount of learner activity data, creating opportunities for personalized learning analytics. However, the complexity, along with the dynamic nature of learners' behavior, makes the segmentation task challenging. This paper presents a comprehensive traditional evaluation and deep learning-based explainable clustering methods for MOOC learner behavioral segmentation. A large-scale dataset of 50,000 learners, described by behavioral features, was used to benchmark several algorithms, including X-means, agglomerative clustering, DBSCAN, spectral clustering, deep embedded clustering, DEC self-supervised clustering, explainable clustering, k-means with prototype interpretation, and meta-learning clustering. The models were assessed using the silhouette score, the Calinski–Harabasz index, and the Davies–Bouldin index. The results prove that X-Means achieved the best overall performance, with a silhouette of 0.512 and CH of 19,604. However, X-Means achieves a DB of 1.081, outperforming both self-supervised learning approaches and deep learning. Understandable clustering showed strong interpretability for competitive metrics: DB is 1.028, and CH is 23,291, highlighting its potential for actionable insights in educational settings. The DBSCAN performed well, capturing impenetrable structures with a silhouette of 0.458, though at a higher noise fraction. In contrast, spectral clustering exhibited poor separation, with a stability silhouette = -0.325 and DB = 6.096, confirming its limitations for high-dimensional behavioral data. Overall, the results suggest that flexible centroid-based methods, particularly X-means, provide reliable and scalable solutions for segmenting MOOC learners, while enabling more effective personalization through targeted interventions in online education.

Keywords: *Clustering algorithms, Educational data mining, Learner segmentation, Massive Open online courses (MOOCs), X-means clustering*

1 INTRODUCTION

Massive open online courses MOOCs create huge amounts of information from millions of students and show complex improvements besides co-enrollment behaviors. These massive-scale datasets allow researchers to distinguish learning patterns and involvement stages besides dropout risks. MOOCs offer a rich delivery of pupil behavior information, taking into consideration huge statistics evaluation, predictive modeling, and personalized coaching. This method

is primarily based on records and has changed online, gaining knowledge of research and practices; however, more attempts are needed to optimize its impact and overcome related issues [1–3].

For education, the segmentation helps create distinct learning pathways based on prior knowledge, students, interaction styles, or educational requirements. Participation and satisfaction, with academic achievement, all rise with strategy. AI-driven segmentation may boost student outcomes with satisfaction metrics by 25%.

Adaptable procedures have varying benefits to different learner segments and demonstrate the significance of segmentation to diverse populations. Segmentation enables the transmission of highly insightful, content-specific recommendations, which are linked to greater learner reflection, perceived benefit, and success. The strategy helps teachers identify and prioritize key quality attributes that different learner groups value, promotes course improvements that relate to particular learner segments, with increased participation and completion rates [4–7]. Modern clustering algorithms such as Deep Embedded Clustering (DEC) and Deep Clustering are becoming increasingly common owing to their ability to handle complex, high-dimensional, and nonlinear data. The techniques employ neural networks to learn feature representations and simultaneously place clusters, thereby improving on previous techniques. Self-supervised with meta-learning models enable clustering methods to adapt to new data distributions as well as workloads with minimum supervision, and hence improve generalization with adaptability. Explainable models allow users to comprehend cluster assignments and model decisions, which is essential for sensitive applications. Noise, which includes unusual values and large-scale data are handled using techniques such as robust self-supervised and network-enhancing [8–11].

The study's objective is to assess various clustering techniques on extensive MOOC behavioral datasets, despite the growing use of machine learning to assess student behaviors. There is a gap in comprehensive data on their performance under the same experimental conditions, as the current study primarily assesses conventional methods or emerging deep learning-based approaches. This study aims to distinguish between algorithmic expertise and practical effectiveness for MOOC scenarios, compare the performance of multiple clustering algorithms on the same dataset, and assess comprehension, flexibility, and accuracy. The goal of a study is to provide a comprehensive understanding of how well these algorithms perform in comparable experimental settings.

2 MOTIVATION

Massive Open Online Courses (MOOCs) have transformed the global learning environment by providing million students with access to excellent education. Nevertheless, it has also made it more challenging for educators to understand and support diverse types of students. Personalized educational strategies and optimal resource

allocation depend on accurate behavioral segmentation. The present research attempts to fill this gap by comparing a number of clustering algorithms, including dynamic splitting, graph-based methods, density-based clustering, hierarchical approaches, deep representation learning, interpretable clustering, and meta-learning-based clustering. The work offers concrete criteria, scaling insights, and support for users and interface designers in developing data-driven, comprehensible, and efficient learner segmentation systems.

3 METHODOLOGY

To achieve powerful, reproducible studies, the current investigation employed a structured approach that included algorithm implementation and evaluation. A large-scale MOOC dataset was preprocessed and normalized before applying a variety of clustering techniques. The following sections cover the dataset used, preprocessing stages, clustering methods, and evaluation metrics.

3.1 Dataset and preprocessing

Dataset: This study analyzes student interaction data from courses between 2012 and 2013 using the open-access MOOC Datasets from HarvardX and MITx, which are archived on Kaggle. The dataset consists of 22 features, including demographic information and course registration data with attendance patterns. These features include the institute, course identity, year, semester, learner age, behavioral engagement characteristics, and course-related information. It scales the behavioral variables that indicate learners' engagement patterns with the platform for clustering analysis to 10. It maintains legal study procedures, all user identifiers are anonymized, and no personally identifying information was stored. The dataset is freely available for academic use and is free of administrative restrictions [12].

Preprocessing Steps: The dataset has been preprocessed to verify that it is suitable for clustering. It underwent numerous processes, including missing-value imputation with parameter scaling and dimension reduction. The median was used to estimate numerical features, and the mode for categorical data. Min-Max scaling modified features, such as high-magnitude characteristics, didn't control distance estimates. The first stage of a process ensures that a dataset is clean, noise-free, and normalized, addressing issues in large-scale MOOC data, such as missing values and heterogeneous data types, as well as feature scaling. Algorithm 1 explains the

preprocessing of the dataset in this scenario.

Algorithm 1: Preprocessing Dataset

Input: Raw MOOC dataset (DataFrame) with 22 attributes, including categorical and numerical features.

Output: Cleaned and standardized DataFrame ready for clustering.

begin

```

Load the raw MOOC dataset (.csv file);
Handle missing values;
    Apply median imputation for numerical features;
    Apply mode imputation for categorical features;
Drop irrelevant columns (e.g., course_id, semester, textual IDs) if not useful for clustering;
Encode categorical variables;
    Convert institute, year, etc., into numerical format using one-hot encoding or label encoding. Remove duplicates to ensure unique learners;
Detect and handle outliers (e.g., using the IQR method or z-score);
Standardize numeric features using StandardScaler, ensuring all features have a mean of 0 and a variance of 1;
Save the preprocessed dataset to a new DataFrame;

```

3.2 Clustering algorithms

Clustering algorithms may be categorized into Traditional Methods, Deep Learning Methods, and Self-Supervised Methods. Traditional methods are competitive with moderate-sized, well-structured data, but deep learning clustering excels at extracting meaningful representations from complex, high-dimensional data. Self-supervised clustering integrates representation learning and clustering, offering enhanced robustness and adaptability, especially in noisy or graph-structured data. Ensemble and hybrid approaches combine strengths across categories, improving robustness while maintaining interpretability. The optimal choice depends on data complexity and scale, as well as interpretability or robustness needs. Recent research highlights the growing power of deep and self-supervised approaches, especially for large, complex datasets [9, 11, 13–15].

To provide a comprehensive comparison, algorithms from three methodological categories were employed:

3.2.1 Traditional methods

Traditional clustering methods are useful for a variety of applications since they are interpretable and efficient. However, they suffer from complicated, high-dimensional, or noisy data, requiring the development of hybrid and adaptive techniques. Centroid-based and hierarchical approaches are often employed in general-purpose clustering, image analysis, and especially pattern recognition. Density-based, along with graph-based approaches, perform well with complicated, noisy, even network-structured data; other than distribution-based methods are more beneficial when the data corresponds to developed statistical distributions [8, 16–18].

A. The X-Means Clustering Algorithm:

The X-Means is an expansion of the commonly used K-Means technique that employs a Bayesian Information Criterion (BIC) to automatically find the best cluster number. This strategy eliminates the need for manual or heuristic K selection and enables iterative cluster splits and enhanced split assessment. X-Means optimizes model fit by balancing complexity and fit to avoid overfitting or underfitting. It maintains a low computational cost of K-Means while introducing just a minor overhead to selecting models and making it suitable for huge datasets. Nevertheless, X-Means is most suitable for data with roughly spherical clusters and may not handle noise or complicated cluster forms as well as more complex algorithms [19]. Algorithm 2 explains how the X-Means Clustering algorithm works in this Scenario.

B. The DBSCAN Clustering Algorithm : Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a technique for clustering points based on density connectivity, identifying flexible forms of clusters as well as distinguishing between noise and outliers. Clusters are characterized as high-density areas separated by regions of low density, based on criteria such as ϵ and MinPts. DBSCAN may discover irregularly shaped clusters without previously defining the number of clusters. It eliminates outliers through designating points beyond concentrated regions like noise. Nevertheless, the success of an algorithm is sensitive to the choice of ϵ and MinPts, making the parameters difficult to select, especially across varying densities [20–22]. Algorithm 3 explains the DBSCAN Clustering Algorithm's work in this Scenario.

Algorithm 2: X-Means Clustering for Behavioral Segmentation**Input:** DataFrame containing standardized learner behavioral features.**Output:** DataFrame with an additional 'cluster' column representing cluster assignments.**begin**

```

Load the preprocessed dataset;
Initialize with K-Means++ using a small
number of clusters (e.g., K=2);
Assign data points to clusters by minimizing
the within-cluster sum of squares;
For each cluster, attempt to split it into two
sub-clusters using K-Means++;
Compute the Bayesian Information Criterion
(BIC) to evaluate the split;
If the split improves BIC, accept the new
clusters; otherwise, keep the original
cluster;
Repeat the splitting process until the
maximum number of clusters (Kmax) is
reached or no further improvement is
observed;
Save the final cluster assignments to the
dataset Generate evaluation metrics
(Silhouette, CH, DB indices)

```

C. The Agglomerative Clustering: Agglomerative clustering is a hierarchical technique that creates a tree-like structure by combining the closest clusters. This procedure divides a tree structure at a chosen level and allows flexible selection of an ideal number of clusters. The tree diagram showing a merging process can be trimmed at various heights to determine the optimal number of clusters. Agglomerative clustering is currently improving in terms of flexibility, community duplicate recognition, and performance indicators such as the Silhouette index [23–26]. Algorithm 4 explains the Agglomerative Clustering Algorithm's work in this Scenario.

3.2.2 Deep learning methods

Deep Embedded Clustering (DEC) is a robust unsupervised learning technique that combines repeated K-Means clustering in a latent space with information and autoencoder-based dimensionality reduction.

Algorithm 3: DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**Input:** DataFrame containing standardized learner behavioral features.**Output:** DataFrame with a 'cluster' column, including a noise label (-1).**begin**

```

Load the preprocessed dataset;
Define two parameters:  $\epsilon$  (neighborhood
radius) and MinPts (minimum number of
points to form a dense region);
For each data point, identify neighbors
within  $\epsilon$  distance;
If neighbors  $\geq$  MinPts, mark as a core point
and form a cluster;
Expand the cluster by including
density-reachable points;
Mark points that do not belong to any cluster
as noise (-1);
Continue until all points are visited;
Save the final cluster assignments into the
dataset;
Evaluate clusters (Silhouette, CH, DB
indices);

```

It enables feature learning with clustering optimization simultaneously. DEC operates through two stages: iterative clustering optimization, in which K-Means clustering improves feature representation via cluster tasks, and autoencoder-based feature learning, in which the autoencoder transforms high-dimensional input data into a lower-dimensional latent space. Its effectiveness in handling massive data sets with non-linear connections has been demonstrated through applications in residential localization, healthcare, and remote sensing [27–29]. Algorithm 5 explains Deep Learning Methods work in this Scenario.

3.2.3 Self-supervised methods

A popular self-supervised technique for learning compact feature representations by carrying out unsupervised clustering combines autoencoders and K-Means. It reduces noise and captures key features of high-dimensional data by compressing it into a lower-dimensional latent space. Following training, an encoded latent vector is subjected to K-Means clustering, which groups related data points according to their learned

representations. Large-scale unlabeled MOOC datasets benefit from this self-supervised clustering technique, which enhances clustering quality through representation learning [30–32]. Algorithm 6 explains Self-Supervised Clustering work in this Scenario.

Algorithm 4: Agglomerative (Hierarchical) Clustering

Input: DataFrame containing standardized learner behavioral features.

Output: DataFrame with a ‘cluster’ column representing hierarchical group assignments.

begin

```

Load the preprocessed dataset;
Treat each learner as a separate cluster
(initial state);
Compute pairwise distances between all
clusters (e.g., Euclidean distance);
Merge the two closest clusters into a new
one;
Update the distance matrix using a linkage
criterion (single, complete, or average
linkage);
Repeat steps 4–6 until the desired number of
clusters is reached;
Assign final cluster labels to the learners;
Save cluster assignments into the dataset;
Evaluate using Silhouette, CH, and DB
indices

```

4 EVALUATION METRICS

For evaluation, the overall clustering results are assessed using a number of internal cluster accuracy indices:

- Clustering quality is determined by calculating a statistic known as a Silhouette Score, which evaluates how well each data point fits into its associated cluster. It is computed by normalizing the average distance to points in the closest other cluster (separation) by the maximum of the two values, and comparing this to the average distance to points in its own cluster (cohesion). A score between -1 and 1 indicates well-clustered points, a score of 0 indicates overlapping clusters, and a score of -1 indicates misclassification [30, 33, 34].

Algorithm 5: Deep Embedded Clustering (DEC)

Input: DataFrame containing standardized learner behavioral features.

Output: DataFrame with clusters derived from learned latent representations.

begin

```

Load the preprocessed dataset;
Train a deep autoencoder to learn
low-dimensional latent features;
Initialize cluster centers using K-Means in
the latent space;
Define a clustering objective function based
on KL divergence;
Iteratively refine cluster assignments and
autoencoder weights jointly;
Converge when the loss function stabilizes;
Assign final cluster labels to learners;
Save the clustered dataset;
Evaluate using clustering metrics;

```

Algorithm 6: Self-Supervised Clustering

Input: DataFrame containing learner behavioral features.

Output: DataFrame with clusters learned from pseudo-labeling.

begin

```

Load the preprocessed dataset;
Generate pseudo-labels using
self-supervised learning (e.g., rotation,
masking);
Train a neural network with pseudo-labels to
learn feature representations;
Extract latent features from the trained
model;
Apply clustering (e.g., hierarchical or
K-Means) on latent features;
Optimize representations and clustering
iteratively;
Assign cluster labels to learners;
Save results with cluster assignments;
Evaluate clusters using Silhouette, CH, and
DB indices;

```

- The Calinski-Harabasz Index compares within-cluster with between-cluster dispersion to deter-

mine clustering quality. Clusters that are dense and well-separated are indicated by higher values. The index is computed by dividing the within-cluster dispersion by the between-cluster dispersion, after accounting for the number of data points and cluster sizes. Clusters with overlapping or poorly defined values are indicated by lower index values, whereas clusters with higher index values are more compact and distinct [30, 35, 36].

- Higher clustering quality is indicated by lower values of the Davies-Bouldin Index (DBI), which calculates an average similarity between clusters. Before averaging a worst-case pair to every cluster, it computes a ratio of within-cluster dispersion to between-cluster separation. Higher DBI values imply overlapping or poorly defined clusters, whereas lower values indicate more compact, well-separated clusters [30, 34, 37].

5 RESULTS ANALYSIS

The clustering experiments were conducted on the dataset containing 50,000 records with 10 numerical features. A variety of algorithms were applied, ranging from classical approaches (X-Means and DBSCAN plus Agglomerative) to advanced methods (Deep Embedded Clustering and Self-Supervised learning with Explainable clustering). Each algorithm was evaluated using three metrics of internal evaluation: silhouette score and Calinski–Harabasz CH index plus Davies–Bouldin DB index.

In this scenario, the best-performing algorithm is X-Means, which achieved the highest silhouette score of 0.512, with a competitive balance between CH (19,604.82) and DB (1.081). This suggests that x-means produced well-detected clusters with reasonable density. DBSCAN ranked second, with a robust silhouette of 0.458, but displayed moderate cluster density (DB = 1.404). It also produced 22 clusters, with approximately 5.3% of the data marked as noise, a typical result for density-based methods.

Among the partitioning methods, also explainable clustering (k-means plus permutation importance) stood out with a solid silhouette score of 0.412 and a highest CH index of 23,291.63, with a lowest DB index of 1.028. This indicates a very good trade-off between cluster cohesion and separation, while also providing interpretability through feature importance analysis. Similarly, agglomerative clustering performed robustly and

produced compact clusters, CH is 22,119.61, and DB is 1.161, with the stable hierarchical structure.

The deep learning approaches presented mixed results. The self-supervised method delivered a silhouette of 0.391 with a reasonably low DB of 1.252, making it competitive but not superior to classical methods. On the other hand, deeply embedded clustering DEC achieved a silhouette of 0.399 but had the highest DB among the viable models (1.752), suggesting fewer compact clusters.

At the lower end, affinity propagation produced a large number of clusters (418), which reduced interpretability as well as stability. Its silhouette score is 0.293, and DB is 1.479 confirmed weaker separation. The spectral clustering failed to capture meaningful structure in the dataset, as reflected in its negative silhouette (- 0.325) and extremely high DB (6.096), making it the worst performer. Fig. 1 below shows the performance comparison of clustering algorithms in our scenario.

The meta-learning component attempted to automatically select the best model agglomerative in this case, with a silhouette $\approx$ of 0.461, but encountered a memory allocation error on a full dataset. Although this prevented a complete evaluation, an initial selection indicates that hierarchical methods remain promising.

As shown in Table 1, it is sorted from the best to the worst performance in this scenario. It showed that X-means had the best silhouette score of 0.512, suggesting well-separated clusters, but spectral clustering had the worst with a negative silhouette score with a large database index.

Table 1 Summary of complexity and properties for the ViViT model's shapes

Rank	Algorithm	Silhouette ↑	Calinski-Harabasz ↑	Davies-Bouldin ↓
1	X-Means	0.512	19,604.82	1.081
2	DBSCAN	0.458	4,883.13	1.404
3	Explainable	0.412	23,291.63	1.028
4	Agglomerative	0.409	22,119.61	1.161
5	Self-Supervised	0.391	19,866.36	1.252
6	Deep Clustering	0.399	13,972.68	1.752
7	Affinity Propagation	0.293	3,612.22	1.479
8	Spectral	-0.325	471.54	6.096

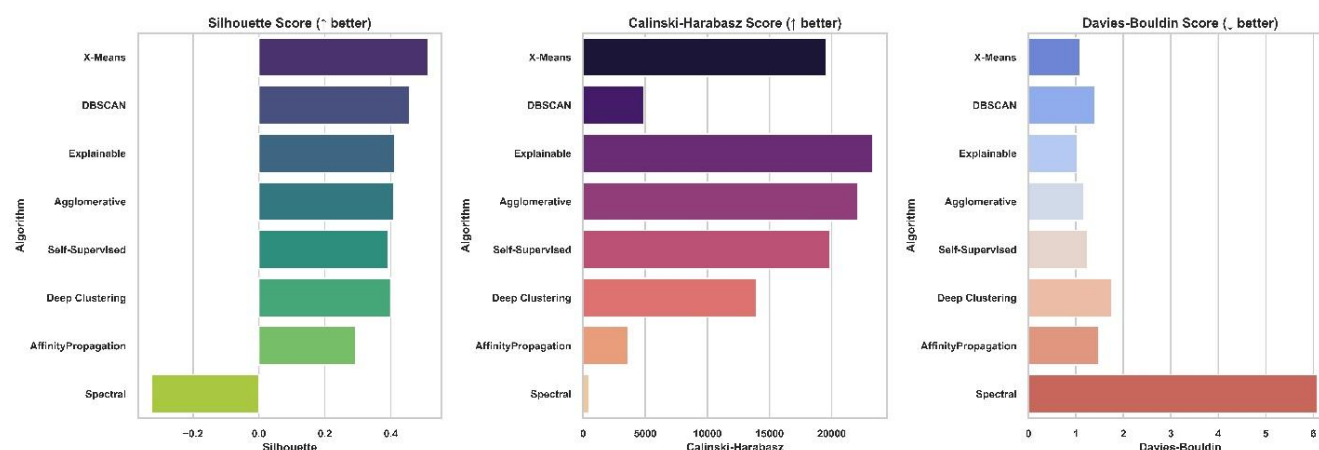


Fig. 1 Performance Comparison of Clustering Algorithms

6 CONCLUSION

This study evaluated diverse clustering algorithms applied to a large-scale dataset of 50,000 MOOC participants, in addition to identifying effective approaches to behavioral segmentation in learning environments. The analysis showed that X-means achieves the best overall performance, with the highest silhouette score and a favorable balance between separation and compactness of the clusters. X-means' success is largely attributed to its flexibility in dynamically selecting an optimal cluster number, which enhances the k-means framework's strengths while mitigating its limitations.

DBSCAN with agglomerative clustering and explainable k-means may show competitive performance, indicating different suitability for segmentation scenarios. The explainable framework, in particular, provides both strong clustering excellence with interpretability over feature importance analysis, which is critical towards actionable educational insights. Spectral and affinity propagation clustering performed poorly and demonstrated limited effectiveness on large-scale behavioral data.

The deep learning-based methods evaluation deep embedded clustering with self-supervised clustering produced comparable results, though less consistent. Nonetheless, these methods hold promise due to their ability to leverage representation learning, but their application at scale was constrained by the need for extensive parameter tuning and higher computational demands. This indicates that, although promising, deep clustering approaches are not yet as practical or efficient as classical clustering methods for large educational datasets.

Finally, a finding confirms that systematic student

segmentation in massive online learning environments is feasible with efficient use of x-means. Its computational efficiency, reliability, and interpretability make it the practical solution for real-world educational analytics. In addition to forward integration of categorical features into hybrid models that syndicate deep neural representations, adaptive clustering methods like x-means, and enhancements to interpretability frameworks, future research should prioritize these priorities. Such improvements will advance both predictive ability and explainability in learner segmentation systems and ultimately support more personalized and effective online education.

Funding source

No funds received.

Data availability

N/A

DECLARATIONS

Conflict of interest

The authors declare no competing interests

Consent to publish

N/A

Ethical approval

N/A

REFERENCES

- [1] Zhou X, Sun A, Zhang J, Lin D. The Crowd in MOOCs: A Study of Learning Patterns at Scale. *Interactive Learning Environments*. 2025;33(3):2136-50. [10.1080/10494820.2024.2392175](https://doi.org/10.1080/10494820.2024.2392175)
- [2] Althibyani HA. Predicting Student Success in MOOCs: A Comprehensive Analysis Using Machine Learning Models. *PeerJ Computer Science*. 2024;10:e2221. [10.7717/peerj-cs.2221](https://doi.org/10.7717/peerj-cs.2221)
- [3] Huang Q, Chen J. Enhancing Academic Performance Prediction with Temporal Graph Networks for Massive Open Online Courses. *Journal of Big Data*. 2024;11(1):52. [10.1186/s40537-024-00918-5](https://doi.org/10.1186/s40537-024-00918-5)
- [4] Al-Kerboly DMA, Hamad MM, Dawood OA. Improving Educational Big Data Based on Web Mining and Association Rules. In: *AIP Conference Proceedings*. vol. 3015. AIP Publishing; 2023. p. 020005. [10.1063/5.0188857](https://doi.org/10.1063/5.0188857)
- [5] Wang R, Ling H, Chen J, Fu H. An Integrated LDA-QFD Approach for Improving Online Course Quality Based on Learners' Reviews. *International Journal of Distance Education Technologies*. 2025;23(1):1-24. [10.4018/IJDET.371203](https://doi.org/10.4018/IJDET.371203)
- [6] Gombert S, Fink A, Giorgashvili T, Jivet I, Di Mitri D, Yau J, et al. From the Automated Assessment of Student Essay Content to Highly Informative Feedback: A Case Study. *International Journal of Artificial Intelligence in Education*. 2024;34(4):1378-416. [10.1007/s40593-023-00387-6](https://doi.org/10.1007/s40593-023-00387-6)
- [7] Naseer F, Khan MN, Tahir M, Addas A, Aejaz SMH. Integrating Deep Learning Techniques for Personalized Learning Pathways in Higher Education. *Heliyon*. 2024;10(11):e32628. [10.1016/j.heliyon.2024.e32628](https://doi.org/10.1016/j.heliyon.2024.e32628)
- [8] Wani AA. Comprehensive Analysis of Clustering Algorithms: Exploring Limitations and Innovative Solutions. *PeerJ Computer Science*. 2024;10:e2286. [10.7717/peerj-cs.2286](https://doi.org/10.7717/peerj-cs.2286)
- [9] AlZuhair MS, Ben Ismail MM, Bchir O. Novel Dual-Constraint-Based Semi-Supervised Deep Clustering Approach. *Sensors*. 2025;25(8):2622. [10.3390/s25082622](https://doi.org/10.3390/s25082622)
- [10] Al-Kerboly DMA, Hamad MM, Dawood OA. Web Mining Algorithms Applied to University Researchers Profiles: A Case Study. *International Journal on Technical and Physical Problems of Engineering*. 2023;15(4):123-33
- [11] Ru Y, Gruninger M, Dou Y. Robust Self Supervised Symmetric Nonnegative Matrix Factorization to the Graph Clustering. *Scientific Reports*. 2025;15(1):7350. [10.1038/s41598-025-92564-x](https://doi.org/10.1038/s41598-025-92564-x)
- [12] Narang K. MOOC Dataset; 2021. Accessed 2026-08-06. Kaggle dataset. Available from: <https://www.kaggle.com/datasets/kanikanarang94/mooc-dataset>
- [13] Goggin SM, Zunder ER. ESCHR: A Hyperparameter-Randomized Ensemble Approach for Robust Clustering across Diverse Datasets. *Genome Biology*. 2024;25(1):242. Peer-reviewed version of the bioRxiv preprint 10.1101/2023.12.18.571953. [10.1186/s13059-024-03386-5](https://doi.org/10.1186/s13059-024-03386-5)
- [14] Sánchez Vines BV, Schubert E, Zimek A, Cordeiro RLF. A Comparative Evaluation of Clustering-Based Outlier Detection. *Data Mining and Knowledge Discovery*. 2025;39(2):13. [10.1007/s10618-024-01086-z](https://doi.org/10.1007/s10618-024-01086-z)
- [15] Al-Kerboly DMA, Hamad MM, Dawood OA. Clustering Algorithms Comparison for University of Anbar Researchers' Google Scholar Profiles. In: *2023 AI-Sadiq International Conference on Communication and Information Technology (AICCIT)*. IEEE; 2023. p. 91-6. [10.1109/AICC-IT57614.2023.10217951](https://doi.org/10.1109/AICC-IT57614.2023.10217951)
- [16] Seong M, Park DK. Hamiltonian Formulations of Centroid-Based Clustering. *Frontiers in Physics*. 2025;13:1544623. [10.3389/fphy.2025.1544623](https://doi.org/10.3389/fphy.2025.1544623)
- [17] Liu Y, Lin X, Chen Y, Cheng R. Multi-Order Graph Clustering with Adaptive Node-Level Weight Learning. *Pattern Recognition*. 2024;156:110843. [10.1016/j.patcog.2024.110843](https://doi.org/10.1016/j.patcog.2024.110843)
- [18] Al-Kerboly DMA, Hamad MM, Dawood OA. Big Data Applications Based on Web Mining Techniques and Recommender Systems: Survey. In: *AIP Conference Proceedings*. vol. 3009. AIP Publishing; 2024. p. 020010. [10.1063/5.0190403](https://doi.org/10.1063/5.0190403)
- [19] Premkumar M, Sinha G, Ramasamy MD, Sahu S, Subramanyam CB, Sowmya R, et al. Augmented Weighted K-Means Grey Wolf Optimizer: An Enhanced Metaheuristic Algorithm for Data Clustering Problems. *Scientific Reports*. 2024;14(1):5434. [10.1038/s41598-024-55619-z](https://doi.org/10.1038/s41598-024-55619-z)
- [20] Li J, Chen F. Gauging Road Safety Advances Using a Hybrid EWM-PROMETHEE II-DBSCAN Model with Machine Learning. *Frontiers in Public Health*. 2024;12:1413031. [10.3389/fpubh.2024.1413031](https://doi.org/10.3389/fpubh.2024.1413031)

- [21] Fang C, Zhou L, Gu X, Liu X, Werner M. A Data Driven Approach to Urban Area Delineation Using Multi Source Geospatial Data. *Scientific Reports*. 2025;15(1):8708. [10.1038/s41598-025-93366-x](https://doi.org/10.1038/s41598-025-93366-x)
- [22] Yang Y, Wang X, Wu X, Lan X, Su T, Guo Y. A Target Detection Algorithm Based on Fusing Radar with a Camera in the Presence of a Fluctuating Signal Intensity. *Remote Sensing*. 2024;16(18):3356. [10.3390/rs16183356](https://doi.org/10.3390/rs16183356)
- [23] Ghorvei M, Karhu T, Hietakoste S, Ferreira-Santos D, Hrubos-Strøm H, Islind AS, et al. A Comparative Analysis of Unsupervised Machine-Learning Methods in PSG-Related Phenotyping. *Journal of Sleep Research*. 2025;34(3):e14349. [10.1111/jsr.14349](https://doi.org/10.1111/jsr.14349)
- [24] Prokop P, Dráždilová P, Platoš J. Overlapping Community Detection in Weighted Networks via Hierarchical Clustering. *PLOS ONE*. 2024;19(10):e0312596. [10.1371/journal.pone.0312596](https://doi.org/10.1371/journal.pone.0312596)
- [25] Sieranoja S, Fränti P. Fast Agglomerative Clustering Using Approximate Traveling Salesman Solutions. *Journal of Big Data*. 2025;12(1):21. [10.1186/s40537-024-01053-x](https://doi.org/10.1186/s40537-024-01053-x)
- [26] Khosravi Y, Hosseinali F, Adresi M. Identifying Accident Prone Areas and Factors Influencing the Severity of Crashes Using Machine Learning and Spatial Analyses. *Scientific Reports*. 2024;14(1):29836. [10.1038/s41598-024-81121-7](https://doi.org/10.1038/s41598-024-81121-7)
- [27] de Kok JWTM, van Rosmalen F, Koeze J, Keus F, van Kuijk SMJ, Castela Forte J, et al. Deep Embedded Clustering Generalisability and Adaptation for Integrating Mixed Datatypes: Two Critical Care Cohorts. *Scientific Reports*. 2024;14(1):1045. [10.1038/s41598-024-51699-z](https://doi.org/10.1038/s41598-024-51699-z)
- [28] Duong PB, Van Herbruggen B, Bröring A, Shahid A, De Poorter E. Error Mitigation for TDoA UWB Indoor Localization Using Unsupervised Machine Learning. *IEEE Sensors Journal*. 2025;25(1):1959-68. [10.1109/JSEN.2024.3496086](https://doi.org/10.1109/JSEN.2024.3496086)
- [29] Priya RS, Vani K. Vegetation Change Detection and Recovery Assessment Based on Post-Fire Satellite Imagery Using Deep Learning. *Scientific Reports*. 2024;14(1):12611. [10.1038/s41598-024-63047-2](https://doi.org/10.1038/s41598-024-63047-2)
- [30] Akgüller Ö, Balcı MA, Cioca G. Clustering Molecules at a Large Scale: Integrating Spectral Geometry with Deep Learning. *Molecules*. 2024;29(16):3902. [10.3390/molecules29163902](https://doi.org/10.3390/molecules29163902)
- [31] Yadalam PK, Natarajan PM, Ardila CM. Variational Graph Autoencoder for Reconstructed Transcriptomic Data Associated with NLRP3-Mediated Pyroptosis in Periodontitis. *Scientific Reports*. 2025;15(1):1962. [10.1038/s41598-025-86455-4](https://doi.org/10.1038/s41598-025-86455-4)
- [32] Nagar S, Farahbakhsh E, Awange J, Chandra R. Remote Sensing Framework for Geological Mapping via Stacked Autoencoders and Clustering. *Advances in Space Research*. 2024;74(10):4502-16. [10.1016/j.asr.2024.09.013](https://doi.org/10.1016/j.asr.2024.09.013)
- [33] Petukhova A, Matos-Carvalho JP, Fachada N. Text Clustering with Large Language Model Embeddings. *International Journal of Cognitive Computing in Engineering*. 2025;6:100-8. [10.1016/j.ijcce.2024.11.004](https://doi.org/10.1016/j.ijcce.2024.11.004)
- [34] Bombina P, Tally D, Abrams ZB, Coombes KR. SillyPutty: Improved Clustering by Optimizing the Silhouette Width. *PLOS ONE*. 2024;19(6):e0300358. [10.1371/journal.pone.0300358](https://doi.org/10.1371/journal.pone.0300358)
- [35] Amin MM, Sani NS, Nasrudin MF, Abdullah S, Chhabra A, Abd Kadir F. Clustering Analysis for Classifying Fake Real Estate Listings. *PeerJ Computer Science*. 2024;10:e2019. [10.7717/peerj-cs.2019](https://doi.org/10.7717/peerj-cs.2019)
- [36] Al-Kerboly DMA, Al-Kerboly DZF. A Comparative Study of Clustering Algorithms for Profiling Researchers in Universities Through Google Scholar. In: 2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI). IEEE; 2024. p. 1-5. [10.1109/ICMI60790.2024.10585953](https://doi.org/10.1109/ICMI60790.2024.10585953)
- [37] Yang D, Wang J, He J, Zhao C. A Clustering Mining Method for Sports Behavior Characteristics of Athletes Based on the Ant Colony Optimization. *Heliyon*. 2024;10(12):e33297. [10.1016/j.heliyon.2024.e33297](https://doi.org/10.1016/j.heliyon.2024.e33297)

How to cite this article

Al-Kerboly DM. A comparative study of traditional, deep, and explainable clustering algorithms for behavioral segmentation of MOOC learners. *Journal of University of Anbar for Pure Science*. 2026; 20(2):320-328. doi:[10.37652/juaps.2025.158955.1369](https://doi.org/10.37652/juaps.2025.158955.1369)