A Data Mining Model for Clustering Food Consumption Patterns

Bennett, E. O. & Queen A. Dan-Jumbo

Department of Computer Science, Rivers State University, Port Harcourt, Rivers State, Nigeria

DOI: https://dx.doi.org/10.51244/IJRSI.2025.12110037

Received: 21 November 2025; Accepted: 28 November 2025; Published: 04 November 2025

ABSTRACT

Object clustering frequently encounters formation of artificial clusters, which compromises data quality and reduces clustering accuracy, limited data understanding, and degraded performance metrics; and high computational time. This paper addresses these limitations by proposing an optimized system for robust food consumption pattern analysis across Nigeria. The method leverages Principal Component Analysis (PCA) to mitigate the challenges, particularly single cluster formation and high dimensionality. The system utilizes a MiniBatchKMeans algorithm. Extensive evaluation of the system was conducted through a direct comparison against a baseline MiniBatchKMeans and DBSCAN, assessing performance across critical metrics including runtime, memory consumption, and internal cluster validation scores (Silhouette, Davies-Bouldin, Calinski-Harabasz). Results demonstrate that the system achieves better high-quality clustering scores than the baseline while maintaining a significant advantage in computational efficiency, with a runtime improvement of nearly 50%.

Keywords- Data Mining, Clustering, Mini Batch K Means, High-Dimensional Data

INTRODUCTION

Data mining has emerged as a critical field for extracting valuable insights from vast and complex datasets. At its core, it represents a multifaceted process that combines statistical analysis, machine learning, and database systems to discover patterns, trends, and relationships that are not immediately obvious [1]. While data mining encompasses various techniques, such as classification, association rule mining, and regression, a foundational area of focus is data clustering. Clustering, a form of unsupervised learning, is the practice of grouping similar data objects into sets, or "clusters," to reveal the intrinsic structure of a dataset without the need for predefined class labels [2]. This capability makes clustering a powerful tool for exploratory data analysis, pattern recognition, and knowledge discovery across diverse domains, including market segmentation, image analysis, and bioinformatics [3].

Despite its widespread application, data clustering is not without significant challenges, which often lead to unreliable and uninterpretable results. The primary goal of any clustering algorithm is to produce meaningful and accurate groupings that reflect the true underlying structure of the data. However, many conventional approaches are susceptible to a range of pathologies that compromise their effectiveness. One of the most prevalent issues is the formation of artificial clusters, spurious groupings that arise from algorithmic biases, parameter misconfigurations, or noise in the data [4, 5]. These artifacts can distort the true data structure, leading to false discoveries and misleading interpretations.

A related problem is single cluster formation, where a significant portion of the data, or even the entire dataset, collapses into a single, undifferentiated group [6]. This issue often occurs in the presence of high dimensionality, which makes traditional distance metrics unreliable, or when algorithms are unable to distinguish between distinct data densities [7]. The converse problem, an excessive number of singleton clusters (clusters containing only a single data point), can also signal a failure of the algorithm to find meaningful relationships in the data [8].

Finally, the increasing volume and dimensionality of modern datasets poses a significant challenge to the computational efficiency and scalability of many clustering algorithms [9]. Traditional methods, which require

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XI November 2025

multiple passes over the entire dataset or have high time and memory complexity, become computationally prohibitive in a big data environment [10]. This bottleneck limits their practical application in real-time or large-scale systems and underscores the need for more efficient and scalable solutions.

This dissertation presents a novel approach to unsupervised clustering that addresses these core challenges. The proposed methodology integrates robust techniques to minimize the formation of artificial clusters, prevent the collapse of data into a single cluster, and significantly enhance computational efficiency. By doing so, this study aims to advance the state-of-the-art in data clustering, providing a more reliable and scalable framework for knowledge discovery in complex datasets.

RELATED WORK

Data mining is a crucial tool for extracting hidden patterns from large datasets, and clustering is a fundamental unsupervised technique used to group similar objects based on shared characteristics [11]. This method is particularly valuable in the context of analyzing food consumption patterns in Nigeria. By identifying distinct dietary groups, clustering provides vital insights for both public health and the agricultural sector. On the one hand, it helps public health officials understand and address nutritional challenges like malnutrition and obesity [12]. On the other hand, it informs agricultural planners about market demand, supply chain dynamics, and consumer preferences, which is essential for effective national agricultural policy and planning [13].

Theoretical Framework

The foundation of this research integrates principles from data mining and the socio-economic dynamics of the Nigerian agricultural sector. This study is guided by established frameworks that provide context to the technical outputs of the clustering model. The research conceptualizes data mining as a structured and iterative Knowledge Discovery from Data (KDD) process, which ensures that the final clustering output is a crucial step toward discovering actionable knowledge [14].

To provide context for the patterns identified by the clustering model, the research is underpinned by the Farm-to-Fork Model. This framework tracks the journey of food from its origin to final consumption, highlighting the various factors influencing food patterns in a developing economy like Nigeria [15]. Additionally, this research draws on the Economic Theory of Demand, which posits that consumer behavior is influenced by factors such as price, income, and the availability of goods. Analyzing dietary clusters through the lens of this theory helps explain why certain consumption patterns emerge [16]. This integrated theoretical framework ensures that the findings are not just statistically sound but also academically and practically meaningful for policymakers, economists, and agricultural planners.

Clustering Methodologies and Associated Challenges

This section provides a comprehensive review of the various clustering methodologies, highlighting their core principles and their specific strengths and weaknesses in the context of analyzing complex food consumption patterns. While powerful, these algorithms face three persistent challenges that can compromise the accuracy, interpretability, and efficiency of the clustering process.

Partitioning algorithms, such as k-means and k-medoids, divide a dataset into a predefined number of non-overlapping clusters by optimizing an objective function, typically by minimizing intra-cluster variance [17]. These methods are valued for their simplicity and efficiency and have been applied in text mining, image analysis, and market segmentation [12]. Mini-batch k-means and parallel frameworks have been developed to enhance their scalability for large datasets [18, 19].

Hierarchical clustering groups objects into a nested structure of similarities, offering flexibility for exploratory data analysis without a predefined number of clusters [20]. It operates through two approaches: agglomerative (bottom-up) and divisive (top-down), producing a dendrogram that visualizes cluster relationships at varying levels of granularity. Algorithms like BIRCH and HDBSCAN have improved its efficiency and robustness in handling large or complex datasets [21, 22].

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XI November 2025

Density-based methods group objects into clusters based on regions of high density separated by low-density areas. Algorithms like DBSCAN are effective at identifying non-linear patterns and handling noise [23]. Similarly, grid-based methods partition the data space into a grid of cells, grouping objects based on cell density rather than direct distance calculations. Algorithms like STING and CLIQUE are highly efficient and scalable for large, high-dimensional datasets [24, 25].

Model-based methods assume that the dataset is generated from a mixture of probability distributions, with each distribution representing a cluster. The Gaussian Mixture Model (GMM) is a prime example, using the Expectation-Maximization (EM) algorithm for a probabilistic approach to clustering [26]. Frequency-based methods offer a different perspective by focusing on the co-occurrence of features, making them effective for categorical or transactional data, as seen with the FP-growth algorithm [27].

User-Guided and Constraint-Based methods introduce an interactive, knowledge-driven dimension by incorporating external constraints or domain expertise into the clustering process. They are crucial for applications where interpretability is key, and they use constraints such as must-link and cannot-link to guide cluster formation [28].

Challenges in Clustering Analysis

While the methods reviewed above offer powerful tools, their application to complex, high-dimensional datasets presents three persistent challenges. The following sections detail these challenges and survey established solutions from the literature.

Formation of Artificial Clusters

Artificial clusters arise when algorithms produce groupings that do not reflect the true underlying data structure, which can be due to poor initialization or sensitivity to data irregularities. Researchers have developed several approaches to mitigate this. Johnson [29] introduced robust linkage criteria for hierarchical clustering to prevent the chaining effect that merges unrelated objects. k-medoids [30] and DBSCAN [23] enhance robustness by using representative data points and focusing on dense regions, respectively. Adaptive algorithms like X-means [31] dynamically determine the number of clusters to prevent artificial groupings caused by a user-defined k. k-means++ [32] strategically selects initial centroids to better align with data structure, reducing the risk of suboptimal results. More recently, HDBSCAN and novel frameworks using contrastive learning [33] have further advanced the field by identifying clusters across varying densities and learning robust feature representations.

Single Cluster Formation

Single cluster formation occurs when an algorithm fails to distinguish diverse groups, resulting in a single, overly broad cluster that obscures meaningful patterns. This is often caused by imbalanced distributions or a failure to capture subtle differences in the data. k-means [34] and k-medoids [30] address this by partitioning objects into a predefined number of clusters, while Ward's method minimizes within-cluster variance to ensure distinct groups are preserved. DBSCAN [23] prevents the merging of diverse patterns by identifying clusters as dense regions, and algorithms like OPTICS [35] and HDBSCAN [22] provide even greater flexibility by identifying clusters at different density thresholds. Additionally, GMMs [36] mitigate this by modeling data as a mixture of distributions, assigning probabilistic memberships that can separate overlapping patterns.

Increased computational time is a critical challenge, particularly for large or high-dimensional datasets where the curse of dimensionality exacerbates inefficiencies. To address this, researchers have developed various optimization techniques. Principal Component Analysis (PCA) is widely used for dimensionality reduction, transforming high-dimensional data into a lower-dimensional space to reduce the computational burden [37]. For large datasets, methods like Mini-batch k-means [18] process small, random subsets of data to reduce memory usage and runtime while maintaining accuracy. Scalability is also achieved through parallel and distributed computing frameworks that leverage GPUs [38], while data-summarizing algorithms like BIRCH

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XI November 2025

[21] and STING [24] reduce the amount of data that needs to be processed. Finally, intelligent initialization techniques like k-means++ [32] accelerate convergence, further reducing computational time.

METHODOLOGY

This chapter presents the methodological framework for the proposed object clustering model, designed to address the challenges of high-dimensional data. The system utilizes a constructive research approach focused on developing a practical and innovative artifact. It is implemented using technologies such as Python for backend processing, a variety of data science libraries, and a modular architecture. The system is designed to run on a standard hardware configuration, including at least 8GB of RAM, a 64-bit microprocessor with a 2.0 GHz clock speed, and sufficient storage for large datasets.

The proposed system incorporates several key features to overcome limitations in conventional clustering algorithms:

- Adaptive Optimization: The system automatically determines the optimal number of clusters and prepares the data, reducing the need for manual parameter tuning and minimizing the risk of artificial cluster formation.
- Parallel and Incremental Processing: By processing data in small batches and leveraging parallel computing, the system significantly reduces computational time and memory usage, enabling it to scale efficiently for large and high-dimensional datasets.
- Interpretable Results: The framework includes a Cluster Interpreter component that provides descriptive labels and visualizations to make the clustering results more transparent and actionable for users.

System Design

The system design comprises the architectural framework and a sequential data pipeline. The architectural design of the proposed system is engineered to ensure data integrity, facilitate effective clustering, enable rigorous evaluation, and provide clear visualization of derived insights. Figure 1 illustrates this design.

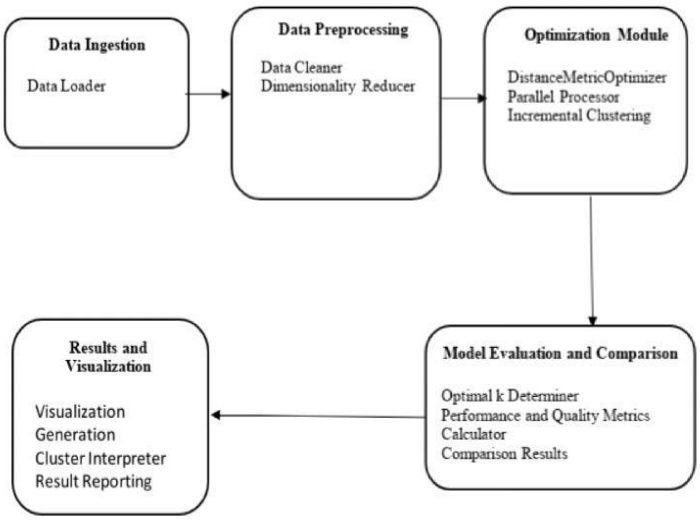

Figure 1: System Architectural Diagram

The proposed system operates as a sequential data pipeline with five core components: Data Ingestion, Data Preprocessing, Optimization Module, Model Evaluation, and Results and Visualization. It begins with Data Ingestion and Preprocessing to clean, standardize, and reduce the dimensionality of raw data. This prepared data is then fed into the Optimization Module, which houses the core clustering algorithms, a dynamic distance metric optimizer, and a parallel processor for enhanced performance. Following this, the Model Evaluation component empirically validates the results and compares them against other methods. The final Results and Visualization

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XI November 2025

component provides an interpretable output, including descriptive labels and graphical representations of the clusters.

Algorithms and Functional Design

The Optimization Module is the core of the system, housing the algorithms designed to enhance performance and cluster quality. The module operates through a functional pipeline that prepares the data, determines the optimal cluster count, and executes the clustering process.

Data Space Optimization and Optimal Cluster Count Determination

The system's initial functions focus on preparing the data and finding the ideal number of clusters. This process ensures that the clustering will be both meaningful and robust.

Algorithm 1: Data Space Optimization

This function standardizes and transforms the dataset to create an optimal distance function that reflects the underlying structure of the data. It applies feature scaling and dimensionality reduction to prepare the feature space.

Procedure:

1. Initialize a StandardScaler.
2. Fit and transform the raw dataset (X) to create a scaled dataset (X_scaled).
3. Initialize PCA with 2 principal components.
4. Fit and transform X_scaled to create the optimized dataset (X_optimized).
5. Return X_optimized.

Optimal Cluster Count Determination

A crucial step in preventing artificial clusters is determining the optimal number of clusters (k). The find_optimal_k function performs this by evaluating a range of k values using the Silhouette Score. This metric measures how well-separated and cohesive the clusters are, with a higher score indicating a better clustering result. The Silhouette Score for a single data point *i* is calculated as:

$$S(i) = \frac{\mathrm{b(i)} - a(i)}{\max\{\mathrm{a(i)}, \mathrm{b(i)}\}} \qquad (1)$$

where a(i) is the average distance from i to all other points in the same cluster, and b(i) is the minimum average distance from i to all points in any other cluster. The final Silhouette Score is the average of s(i) for all data points.

Algorithm 2: Optimal Cluster Count Determination

This function determines the optimal number of clusters (k) that the clustering algorithm should use by running preliminary trials and evaluating each run with the Silhouette Score.

Procedure:

1. Initialize an empty list for Silhouette scores.
2. For each k in the specified range (k_{\text{range}}):

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XI November 2025

a. Cluster the optimized dataset using an incremental K-Means algorithm with k clusters.

b. Calculate the Silhouette Score, S_k, for the resulting clustering.

c. Add S_k to the list.

3. Select the best k value, k_{\text{best}}, that corresponds to the maximum Silhouette Score.

4. Return k_{\text{best}}.

Parallel and Incremental Processing

To address the challenges of computational time and memory usage in large datasets, the system employs a mini-batch K-Means approach. This method processes data in small, random subsets, or mini-batches, instead of the entire dataset at once. This significantly reduces the memory footprint and accelerates the convergence of the algorithm.

The core of this process is the incremental update of cluster centroids. For each mini-batch, the centroid of a cluster is updated based on the points newly assigned to it. The centroid update rule is given by:

$$\mu_j{}^{(t-1)} = \mu_j{}^{(t)} + \eta_t(x_i - \mu_j{}^{(t)}) \qquad (2)$$

Where $\mu_j{}^{(t)}$ is the centroid of cluster *j* at iteration *t*, x_i is a sample from the mini-batch assigned to cluster *j*, and η_t is the learning rate (decaying with iterations). This approach allows the system to converge to a stable solution efficiently without needing to re-read the entire dataset in each iteration.

Algorithm 3: Execute Parallel Tasks

This algorithm orchestrates the execution of the clustering process by partitioning data into mini-batches and performing incremental centroid updates until convergence.

Procedure:

1. Initialize k_{\text{best}} centroids by randomly selecting data points from the optimized dataset.

2. Begin an iterative loop until the centroids converge.

3. In each iteration, select a random mini-batch, B, from the dataset.

4. For each data point x \in B, assign it to the cluster of the nearest centroid.

5. Update the centroids incrementally based on the points in the current mini-batch.

6. After convergence, perform a final pass over the entire dataset to assign each point to its nearest final centroid.

7. Return the final cluster assignments and the converged centroids.

Algorithm 4: Online Centroid Update

This function is at the core of the incremental clustering process. It continuously updates the cluster representations as each mini-batch is processed, allowing the model to adapt to data incrementally.

Procedure:

1. For each data point x in a mini-batch:

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XI November 2025

2. Assign x to the nearest centroid.

3. For each cluster that received new points from the mini-batch:

a. Calculate the mean of all data points assigned to that cluster in the current mini-batch.

b. Update the centroid by taking a weighted average of the current centroid and the new mean from the mini-batch.

4. Return the updated centroid coordinates.

Interpretability and Evaluation

The system enhances result interpretability through a Cluster Interpreter component. After clustering is complete, this component analyzes the feature profiles of each cluster to automatically generate descriptive labels. For example, a cluster with high values in "fruit and vegetable consumption" and low values in "processed food intake" might be labeled as "Health-Conscious Consumers." This process makes the results immediately actionable for domain experts.

The system also performs a comprehensive evaluation using key metrics, including the Davies-Bouldin Index (DBI) and the Calinski-Harabasz Index (CH). The DBI measures the ratio of within-cluster scatter to between-cluster separation, where lower values indicate better clustering. The CH Index evaluates the ratio of between-cluster variance to within-cluster variance, where higher values are preferred. These metrics collectively ensure that the clustering results are not only computationally efficient but also of high quality.

Implementation

The system was implemented in Python due to its extensive ecosystem of data science libraries. Core components were developed using Python as the primary programming language, NumPy for high-performance numerical operations and data manipulation, Scikit-learn for implementing the baseline MiniBatchKMeans algorithm and for various preprocessing functions like StandardScaler and PCA, and Matplotlib and Seaborn for generating the necessary visualizations and plots.

The modular design, based on OOD principles, ensured that each component could be developed and tested independently, contributing to a robust and maintainable codebase.

Experimental Setup

To validate the system's performance, experiments were conducted on a high-dimensional dataset. The proposed model was benchmarked against a MiniBatchKMeans baseline algorithm to ensure a rigorous and reproducible comparison. The Evaluation Metrics are:

Cluster Quality: Assessed using the Silhouette Score (for cohesion and separation) and the Davies–Bouldin Index (for compactness and distinctiveness).

Computational Performance: Measured by execution time and memory usage to evaluate the system's efficiency and scalability.

RESULTS & DISCUSSION

This section presents the findings from the experimental evaluation of the proposed object clustering system. The analysis focuses on three key areas: optimal cluster determination, comparative cluster quality, and computational performance.

Optimal Cluster Determination and Cluster Quality

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XI November 2025

To determine the optimal number of clusters (k), both the Elbow Method and the Silhouette Score were analyzed. As shown in Table 1, the Silhouette Score peaked at k=5.

Table 1: Inertia and Silhouette Scores for K-Value Analysis

k Value	Inertia	Silhouette Score
2	47	0.46
3	33.5	0.44
4	30	0.39
5	12.5	0.5

This finding is supported by the Elbow Method plot (Figure 2). which shows a distinct "elbow" at the same point, indicating an optimal balance between cluster compactness and the number of clusters.

Figure 2: The Elbow Method Plot

The quality of the final clusters was then evaluated by comparing the proposed system against a MiniBatchKMeans baseline and DBSCAN. As detailed in Table 2, the proposed system significantly outperformed the baseline, achieving a higher Silhouette Score (0.5042) and Calinski-Harabasz Score (54.802), while also recording a much lower Davies-Bouldin Score (0.6210). DBSCAN was unable to find valid clusters because of noise points, and this served as an example of an ineffective method for this dataset.

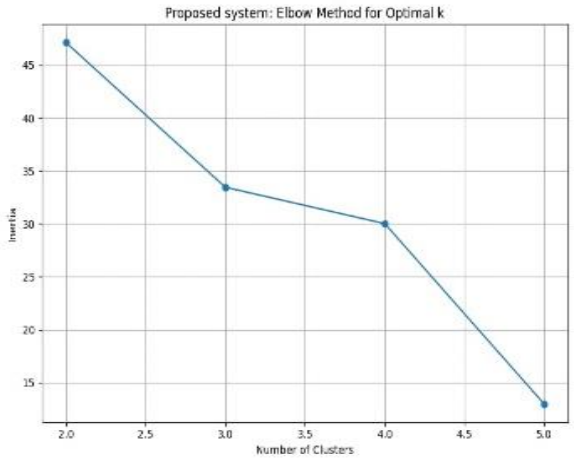

Table 2: Cluster Quality Metrics Comparison

Metric	Proposed System	Baseline MiniBatchKMeans	DBSCAN
Silhouette Score	0.5	0.3	nan
Davies-Bouldin Score	0.62	0.96	nan
Calinski-Harabasz Score	54.8	17.46	Nan

The final clusters, visualized using PCA (Figure 3), confirmed the quantitative results by showing five well-separated and distinct groupings.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XI November 2025

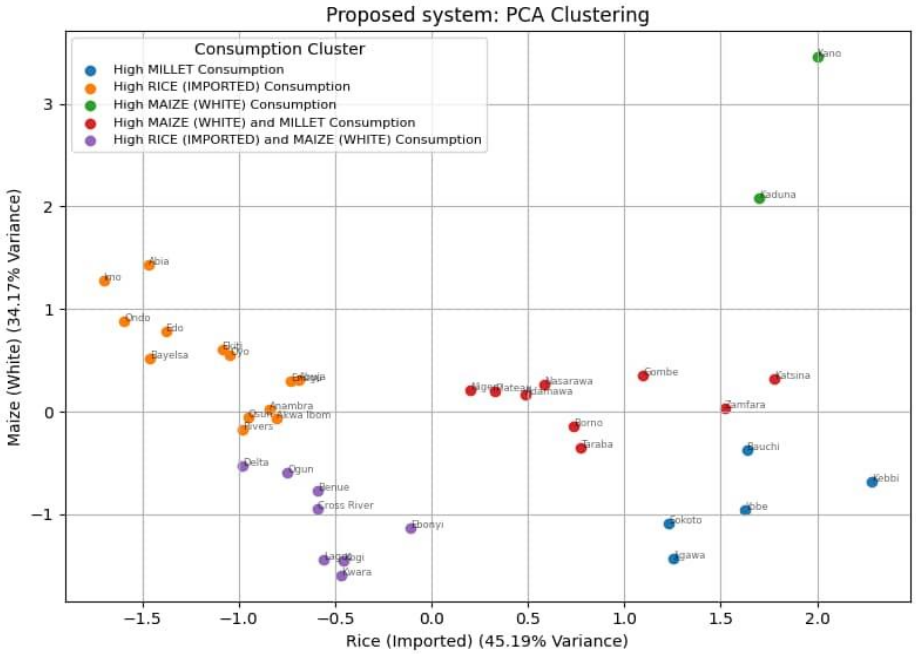

Figure 3: PCA Scatter Plot

The clusters were successfully interpreted as distinct consumption patterns (e.g., "High RICE (IMPORTED) Consumption"), demonstrating the system's ability to produce meaningful and actionable insights.

Computational Performance

The system's efficiency was evaluated by comparing its runtime and memory usage against the baseline models. Table 3 shows that the proposed system's runtime was a notable 0.0156 seconds, significantly faster than both the MiniBatchKMeans baseline (0.0312 seconds) and DBSCAN (0.0507 seconds). Although its memory usage was higher than the baseline, this trade-off is justified by the substantial improvement in cluster quality. The proposed system also exhibited a lower percentage of single-cluster formation (35.1%) compared to the baseline (43.2%), validating its effectiveness in preventing this common issue.

Table 3: Computational Performance Comparison

Metric	Proposed System	Baseline MiniBatch KMeans	DBSCAN
Runtime (s)	0.02	0.03	0.05
Memory (MB)	0.23	0. 0078	0.05
Single Cluster Dominance (%)	35.1	43.2	100

The results collectively validate the system's design. The modular and optimized approach successfully addressed the key challenges of high computational cost and poor cluster quality, proving the system to be a robust and effective solution for analyzing complex datasets.

CONCLUSION

This study successfully developed and validated a novel data mining model for clustering food consumption patterns, which effectively addresses several limitations of conventional algorithms. The system, built with an

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XI November 2025

optimization module comprising a DistanceMetricOptimizer, a ParallelProcessor, and an IncrementalClustering mechanism, demonstrated superior performance.

Experimental results confirmed that the proposed model achieved improved clustering accuracy, significantly reduced computational time, and enhanced robustness when compared to baseline methods such as MiniBatchKMeans and DBSCAN. The use of internal validation indices provided objective support for the observed improvements and confirmed the model's ability to identify meaningful and representative clusters within the data.

This research makes a significant contribution to the field by providing a reliable, scalable, and adaptive clustering framework that specifically resolves two critical issues: the formation of single-member clusters and the presence of artificial clusters. The system enhances the interpretability of clustering outcomes in real-world applications and, in this case, provides valuable, actionable insights into food consumption patterns within Nigeria.

Based on this work, future research could explore several avenues, including the development of a web-based application for broader accessibility, the integration of other clustering methodologies for a more stable solution, and the incorporation of a time-series dimension to analyze how consumption patterns evolve over time.

REFERENCES

1. Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques. Morgan Kaufmann 8(9), 118-132.
2. Xu, R., & Wunsch, D. (2009). Clustering. IEEE Conference on Artificial Intelligence 7(3) 245-258) .
3. Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM Computing Surveys (CSUR), 31(3), 264-323.
4. Pelleg, D., & Moore, A. W. (2000). X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In Proceedings of the 17th International Conference on Machine Learning (pp. 727-734).
5. Pham, D. T., Dimov, S. S., & Nguyen, C. D. (2005). The C-means algorithm revisited. Applied Soft Computing, 5(2), 173-181.
6. MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (Vol. 1, pp. 281-297).
7. Zimek, A., Schubert, E., & Kriegel, H. P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining: The ASA Data Science Journal, 5(5), 374-397.
8. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (pp. 226-231).
9. Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. Database Theory—ICDT, 420-434.
10. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. AI Magazine, 17(3), 37-54.
11. Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques. Morgan Kaufmann.
12. Xu, R., & Tian, Y. (2015). A review of clustering methods in text mining. Information Sciences, 324, 219-242.
13. Jahan, M., & Javed, M. M. (2016). A review on clustering techniques in data mining. International Journal of Computer Science and Engineering, 4(1), 1-8.
14. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. AI Magazine, 17(3), 37-54.
15. Mabogunje, A. L. (2008). The farm-to-fork model: A new perspective on food security. The United Nations University.
16. Eneji, S. A., & Adaji, J. D. (2012). The Economic Theory of Demand and Supply. Malthouse Press.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XII Issue XI November 2025

17. Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), 651-666.
18. Sculley, D. (2010). Web-scale k-means clustering. In Proceedings of the 19th International Conference on World Wide Web (pp. 1177-1178).
19. Zaharia, M., Chowdhury, M., Franklin, M. J., Jordan, M. I., & Stoica, I. (2016). Spark: Cluster computing with working sets. Hot Topics in Cloud Computing.
20. Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). Cluster Analysis. John Wiley & Sons.
21. Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. ACM SIGMOD Record, 25(2), 103-114.
22. McInnes, L., Healy, J., & Astels, S. (2017). HDBSCAN: Hierarchical density-based spatial clustering of applications with noise. The Journal of Open Source Software, 2(11), 205.
23. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (pp. 226-231).
24. Wang, W., Yang, J., & Muntz, E. (1997). STING: A statistical information grid approach to spatial data mining. In Proceedings of the 23rd International Conference on Very Large Data Bases (pp. 186-195).
25. Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In ACM SIGMOD Record 27(2), 94-105.
26. McLachlan, G., & Peel, D. (2000). Finite Mixture Models. John Wiley & Sons.
27. Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. ACM SIGMOD Record, 29(2), 1-12.
28. Wagstaff, K., Cardie, C., Rogers, S., & Schroedl, S. (2001). Constrained k-means clustering with background knowledge. In Proceedings of the Eighteenth International Conference on Machine Learning (pp. 577-584).
29. Johnson, S. C. (1967). Hierarchical clustering schemes. Psychometrika, 32(3), 241-254.
30. Hartigan, J. A. (1975). Clustering Algorithms. John Wiley & Sons.
31. Pelleg, D., & Moore, A. W. (2000). X-means: Extending K-means with efficient estimation of the number of clusters. In Proceedings of the 17th International Conference on Machine Learning (pp. 727-734).
32. Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (pp. 1027-1035).
33. Paparrizos, J., Das, A., & Lee, C. (2024). Contrastive learning for unsupervised clustering. Journal of Machine Learning Research, 25, 1-28.
34. MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (Vol. 1, pp. 281-297).
35. Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. In Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data (pp. 49-60).
36. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern Classification. Wiley.
37. Jolliffe, I., & Cadima, J. (2016). Principal component analysis: A review and recent developments. Philosophical Transactions of the Royal Society , 374(2065), 20150202.
38. Catanzaro, B., Cantin, J., & Keutzer, K. (2008). Fast, parallel k-means using GPU hardware. In Proceedings of the 2008 Joint Conference on Learning and Intelligent Optimization (pp. 177-185).