International Journal of Multidisciplinary Research and Publications
ISSN (Online): 2581-6187

Quantitative Analysis of the Spatial Impact of Data Normalization on the Fuzzy C-Means (FCM) Algorithm

Cibamba Kanyinda Jean[1], Kalonzo Musenga Trésor[2], Aluba MULAMBA Daniel[3], Kutumbakana Mavanga Serge[4]
[1]Research assistant at the Academy of Fine Arts
[2]chef de travaux à Isp- Popokabaka/kwango
[3]Assistant à isp-Baraka
[4]Assistant à UPN-Kinshasa

Abstract*—This article examines the critical impact of data preprocessing on the performance and geometric relevance of the Fuzzy C-Means (FCM) fuzzy partitioning algorithm. Intrinsically based on minimizing a quadratic cost function using Euclidean distance, the FCM model exhibits severe vulnerability to inter-variable scaling disparities. Through a rigorous numerical simulation modeling ten taxpayers assessed on three asymmetric tax dimensions (Turnover, Import-Export, and Royalties), complemented by an extension to a synthetic dataset of 100 taxpayers, we empirically demonstrate how omitting normalization leads to a monopolistic capture of variance by high-magnitude variables, masking critical signals carried by low-magnitude variables. The in-depth comparative analysis of fuzzy membership matrix structures U, internal validity indices (PC, PE), and external consistency and geometric separability metrics (Silhouette and Davies-Bouldin indices), complemented by a study of convergence trajectories, mathematically validates the absolute necessity of scaling to preserve the metric isotropy of the classification space*.

***Keywords*—** *Fuzzy C-Means, Min-Max Normalization, Euclidean Distance, Geometric Distortion, Silhouette Index, Davies-Bouldin Index, Algorithmic Convergence, Tax Compliance Audit, FCM-NASH Model.*

I. Introduction

Applying data normalization upstream of the Fuzzy C-Means (FCM) model is widely accepted in contemporary scientific literature. Since the FCM model is intrinsically based on calculating geometric distances (generally Euclidean distance), it proves extremely sensitive to even the slightest differences in scale between explanatory variables. When assessing conformity or profiling economic entities, analysts frequently handle indicators of heterogeneous natures and magnitudes. For example, the revenue of a large company is expressed in orders of magnitude significantly higher than its operating royalties, creating a de facto structural imbalance during direct classification. This article presents a rigorous synthesis of the scientific arguments in favor of this standardization, supported by the mathematical logic of major publications in the field, and corroborates these foundations with an explicit computer simulation incorporating a comprehensive comparative study (raw data versus standardized data) introducing the Silhouette index and the Davies-Bouldin index.

II. Mathematical Foundations and Bias of the FCM Algorithm

Since the FCM model is intrinsically based on the calculation of geometric distances, it proves extremely sensitive to differences in scale between the variables in the description space.

To understand the sensitivity of the FCM model to scale differences, it is necessary to define the objective cost function J m that the FCM algorithm seeks to minimize: $J_m(U,V) = \sum_{i=1}^{N}\sum_{j=1}^{C} {u_{ij}}^m \|X_i - V_j\|^2 =$

Où :

- N is the total number of observations.
- C is the number of predefined clusters.
- u_ij∈[0,1] represents the degree of fuzzy membership of observation X_i in cluster j.
- m is the fuzzy exponent (m > 1).
- V_j is the centroid of cluster j.
- ||X_i - V_j|| denotes the standard similarity metric, generally defined by the Euclidean distance.

The lack of normalization of the variables used in calculating this Euclidean distance leads to two major biases identified in research:

A. The absolute dominance of high-variance variables: If variable A varies within a wide range (e.g., from 0 to 10,000) while variable B varies within a narrow range (e.g., from 1 to 5), variable A will monopolize almost the entire calculation of the root mean square distance. Consequently, the algorithm will almost systematically ignore the analytical contribution and informative variance provided by variable B.

B. Structural deformation of geometric space: The FCM algorithm implicitly postulates the existence of spherically morphological clusters within the vector space. The introduction of highly disparate scales disproportionately stretches the coordinate axes, forcing the model to search for topological structures in a deformed, ellipsoidal-type frame, which drastically degrades the separability of fuzzy partitions.

III. Algorithmic Impacts Documented in the Literature

Empirical and theoretical work dedicated to data preprocessing for fuzzy partitioning highlights three major benefits of enabling normalization:

Cibamba Kanyinda Jean, Kalonzo Musenga Trésor, Aluba MULAMBA Daniel, and Kutumbakana Mavanga Serge, "Quantitative Analysis of the Spatial Impact of Data Normalization on the Fuzzy C-Means (FCM) Algorithm," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 9, Issue 3, pp. 17-23, August 2026.

A. Optimization of Convergence Speed and Reduction of Computation Time

Normalization induces a more stable gradient descent during the iterative phases of updating the membership matrix (U) and centroid vectors (V). Comparative analyses of the literature demonstrate that the time required for the FCM algorithm to reach its stopping criterion (defined by a tolerance threshold ε) decreases drastically after scaling. By standardizing the variances, we prevent the error gradients from undergoing violent oscillations that slow convergence.

B. Mitigating Sensitivity to Outliers

Implementing appropriate scaling techniques mitigates the disproportionate influence of outliers. Without normalization, a single outlier positioned on a large-scale variable has sufficient geometric leverage to artificially shift the centroid of a fuzzy cluster, thus biasing the semantics of the partition.

C. Improving Cluster Validity Indices

The mathematical quality of fuzzy partitions is traditionally assessed using criteria such as the Partition Coefficient (PC) or Partition Entropy (PE). Scientific publications consistently demonstrate a maximization of PC scores (tending towards 1) and a minimization of PE scores (tending towards 0) when the data have been previously normalized. This confirms that the generated fuzzy boundaries are sharper, more consistent, and more faithful to the actual topology of the multivariate data.

Typology of Recommended Normalization Techniques

The choice of preprocessing method should be dictated by the statistical properties of the dataset. The following table summarizes the approaches validated by prior research on FCM:

Method	Mathematical formulation	Recommended application context in the literature
Min-Max (Rescaling)	$x' = \frac{(x - x\_min)}{(x\_max - x\_min)}$	Ideal when the theoretical data boundaries are fixed and no major outliers are present. Strictly bounds the space between [0, 1].
Z-score (Standardisation)	$x' = \frac{(x - \mu)}{\sigma}$	Recommended when the data follows a normal distribution or contains a few outliers, as it preserves the shape of the distribution while equalizing the variance ($\sigma^2 = 1$).
Normalization by sum	$x' = \frac{x}{\sum x}$	Frequently used in text processing (word frequency) or gene expression analysis (microarrays) to analyze relative proportions rather than absolute values.

Additional note: In the face of measurement noise or highly skewed distributions, the literature recommends the use of robust variants, such as the Z-score based on the median and the median absolute deviation (MAD).

Presentation of External Validity Indices: Silhouette and Davies-Bouldin

To rigorously validate the geometric organization of the clusters, the analysis incorporates two fundamental metric indices calculated after hardening the membership matrix U:

- Silhouette Index (SIL): This index assesses the fit of a sample to its cluster relative to other clusters. For a sample i, where a(i) represents the average intra-cluster distance and b(i) the average distance to the nearest cluster. The average value, between -1 and +1, indicates strong clustering when it is close to +1.
- Davies-Bouldin Index (DB): This indicator measures the average similarity between each cluster and its most similar cluster. It is defined as the average across all clusters of the maximum ratio of the sum of intra-cluster dispersions divided by the inter-centroid distance. Mathematically, , where S is the internal dispersion of cluster i and i=1..C j≠i

= ‖ ‖ - ‖ the distance between centroids i and j. Unlike the Silhouette, a lower DB index (tending towards ij) proves a higher quality partitioning, signifying dense and highly isolated clusters.

Experimental Framework and Enhanced Python Script (Expanded Dataset N=100)

To strengthen the statistical evidence on a sample more representative of the realities of multi-agency tax audits (DGI, DGDA, DGRAD in the DRC), we present a script simulating a universe of 100 taxpayers divided into three distinct profiles (Large Enterprises, Medium-Sized Enterprises, and Asymmetric Profiles with a High Risk of Fraud)

PYSPARK SCRIPT TO GENERATE A COMPARISON BETWEEN STANDARDIZED AND RAW DATA

```
!pip install scikit-fuzzy
Collecting scikit-fuzzy
  Downloading scikit_fuzzy-0.5.0-py2.py3-none-any.whl.metadata (2.6 kB)
Downloading scikit_fuzzy-0.5.0-py2.py3-none-any.whl (920 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
━━━━━━━━ 920.8/920.8 kB 24.3 MB/s eta 0:00:00
Installing collected packages: scikit-fuzzy
Successfully installed scikit-fuzzy-0.5.0 [12]
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pyspark.sql import SparkSession
from pyspark.ml.feature import VectorAssembler, MinMaxScaler
#
# ============================================
# 1. SPARK SESSION INITIATION
#
# ============================================
spark = SparkSession.builder \
    .appName("FCM_Comparatif_Normalisation_RDC") \
    .config("spark.sql.execution.arrow.pyspark.enabled", "true") \
    .getOrCreate()
#
# ============================================
# 2. GENERATION OF SYNTHETIC DATA (25,000 TAXPAYERS)
```


Cibamba Kanyinda Jean, Kalonzo Musenga Trésor, Aluba MULAMBA Daniel, and Kutumbakana Mavanga Serge, "Quantitative Analysis of the Spatial Impact of Data Normalization on the Fuzzy C-Means (FCM) Algorithm," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 9, Issue 3, pp. 17-23, August 2026.

```
#
==========================================
np.random.seed(42)
n_rows = 25000

# X1 : Taux de variation des déclarations (%)
taux_variation = np.random.uniform(-50, 100, n_rows)
# X2 : Volume des recettes déclarées en Francs Congolais
(CDF)
recettes_cdf = np.random.exponential(scale=500000000,
size=n_rows) + 15000000
data = [(float(t), float(r)) for t, r in zip(taux_variation,
recettes_cdf)]
df = spark.createDataFrame(data, ["Taux_Variation_Pct",
"Recettes_CDF"])
# Assemblage et Normalisation Min-Max
assembler =
VectorAssembler(inputCols=["Taux_Variation_Pct",
"Recettes_CDF"], outputCol="features")
df_vector = assembler.transform(df)
scaler = MinMaxScaler(inputCol="features",
outputCol="scaled_features")
scaler_model = scaler.fit(df_vector)
df_normalized = scaler_model.transform(df_vector)
min_vals = scaler_model.originalMin.toArray()
max_vals = scaler_model.originalMax.toArray()
# Extraction des matrices NumPy
pdf = df_normalized.select("Taux_Variation_Pct",
"Recettes_CDF", "scaled_features").toPandas()
X_brut = pdf[["Taux_Variation_Pct",
"Recettes_CDF"]].values
X_scaled = np.array([v.toArray() for v in
pdf["scaled_features"]])
#
==========================================
# 3. FUZZY C-MEANS ALGORITHM & CRISP
VALIDATION INDICES
#
==========================================
def fuzzy_c_means(X, c=3, m=2, max_iter=100, error=1e-5):
    n_samples, n_features = X.shape
    U = np.random.rand(n_samples, c)
    U = U / np.sum(U, axis=1, keepdims=True)
    for _ in range(max_iter):
        U_old = U.copy()
        Um = U ** m
        centroids = (Um.T @ X) / np.sum(Um, axis=0,
keepdims=True).T
        dists = np.linalg.norm(X[:, np.newaxis] - centroids,
axis=2)
        dists = np.fmax(dists, 1e-10)
        inv_dists = (1.0 / dists) ** (2 / (m - 1))
        U = inv_dists / np.sum(inv_dists, axis=1,
keepdims=True)
        if np.linalg.norm(U - U_old) < error:
            break
    return centroids, U
def compute_validation_indices(X, labels, centroids):
    """Calcule Silhouette et Davies-Bouldin de manière
vectorisée."""
    n_samples = X.shape[0]
    c = len(centroids)
    # 1. Davies-Bouldin
    S = np.zeros(c)
    for i in range(c):
        cluster_pts = X[labels == i]
    if len(cluster_pts) > 0:
            S[i] = np.mean(np.linalg.norm(cluster_pts -
centroids[i], axis=1))
    R = np.zeros((c, c))
    for i in range(c):
        for j in range(c):
            if i != j:
                denom = np.linalg.norm(centroids[i] - centroids[j])
                R[i, j] = (S[i] + S[j]) / denom if denom > 0 else 0
    db_index = np.mean(np.max(R, axis=1))
    # 2. Silhouette (2000-point sample for optimal
performance)
    idx = np.random.choice(n_samples, min(2000, n_samples),
replace=False)
    X_sub = X[idx]
    labels_sub = labels[idx]
    silhouettes = []
    for i, pt in enumerate(X_sub):
        l_pt = labels_sub[i]
        same_cluster = X_sub[labels_sub == l_pt]
        if len(same_cluster) > 1:
            a = np.mean(np.linalg.norm(same_cluster - pt, axis=1))
        else:
            a = 0
        b = float('inf')
        for k in range(c):
            if k != l_pt:
                other_cluster = X_sub[labels_sub == k]
                if len(other_cluster) > 0:
                    dist_b = np.mean(np.linalg.norm(other_cluster -
pt, axis=1))
                    b = min(b, dist_b)
        if a == 0 and b == float('inf'):
            silhouettes.append(0)
        else:
            silhouettes.append((b - a) / max(a, b))
    return np.mean(silhouettes), db_index
#
==========================================
# 4. COMPARATIVE ANALYSIS EXECUTION
#
==========================================
c_clusters = 3
# --- CAS 1 : RAW DATA ---
centroids_b, U_b = fuzzy_c_means(X_brut, c=c_clusters)
labels_b = np.argmax(U_b, axis=1)
4
sil_b, db_b = compute_validation_indices(X_brut, labels_b,
centroids_b)
# --- CAS 2 : STANDARDIZED DATA ---
```


Cibamba Kanyinda Jean, Kalonzo Musenga Trésor, Aluba MULAMBA Daniel, and Kutumbakana Mavanga Serge, "Quantitative Analysis of the Spatial Impact of Data Normalization on the Fuzzy C-Means (FCM) Algorithm," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 9, Issue 3, pp. 17-23, August 2026.

```
entroids_s, U_s = fuzzy_c_means(X_scaled, c=c_clusters)
labels_s = np.argmax(U_s, axis=1)
# Calcul des indices dans l'espace géométrique normalisé où l'algorithme a travaillé
sil_s, db_s = compute_validation_indices(X_scaled, labels_s, centroids_s)
# Rétablissement des centroïdes à l'échelle réelle pour la table finale
centroids_s_reels = centroids_s * (max_vals - min_vals) + min_vals

# Add results to dataframe for extraction
for i in range(c_clusters):
    pdf[f'Brut_U{i+1}'] = U_b[:, i]
    pdf[f'Norm_U{i+1}'] = U_s[:, i]
pdf['Class_Brut'] = labels_b + 1
pdf['Class_Norm'] = labels_s + 1

#
# 5. PRESENTATION OF RESULTS
#
print("\n" + "="*85)
print("   TABLEAU COMPARATIF DES INDICES DE VALIDATION")
print("="*85)
print(f"Indice de Silhouette (Proche de 1)  | Brute: {sil_b:.4f} | Normalisée: {sil_s:.4f}")
print(f"Indice de Davies-Bouldin (Proche de 0)| Brute: {db_b:.4f}       | Normalisée: {db_s:.4f}")
print("="*85)

print("\n" + "="*85)
print("   COORDONNÉES COMPARATIVES DES CENTROÏDES (ÉCHELLE RÉELLE)")
print("="*85)
for i in range(c_clusters):
    print(f"Profil Type {i+1} [SANS Normalisation] : Taux = {centroids_b[i,0]:.2f}% | Recettes = {centroids_b[i,1]:,.0f} CDF")
    print(f"Profil Type {i+1} [AVEC Normalisation] : Taux = {centroids_s_reels[i,0]:.2f}% | Recettes = {centroids_s_reels[i,1]:,.0f} CDF")
    print("-" * 85)

print("\n=== EXTRAIT DE LA MATRICE D'APPARTENANCES DES 20 PREMIERS CONTRIBUABLES ===")
cols = ["Taux_Variation_Pct", "Recettes_CDF", "Brut_U1", "Norm_U1", "Brut_U2", "Norm_U2", "Class_Brut", "Class_Norm"]
print(pdf[cols].head(20).to_string(index=True, formatters={
    'Taux_Variation_Pct': '{:,.2f}%'.format, \
    'Recettes_CDF': '{:,.0f} CDF'.format, \
    'Brut_U1': '{:.3f}'.format, 'Norm_U1': '{:.3f}'.format, \
    'Brut_U2': '{:.3f}'.format, 'Norm_U2': '{:.3f}'.format
}))

#
# 6. VISUALIZATION OF THE TWO SCENARIOS
#
sample_pdf = pdf.sample(n=2500, random_state=42)
colors = {1: '#e41a1c', 2: '#377eb8', 3: '#4daf4a'}
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 6))

# Graphique 1 : Données Brutes
ax1.scatter(sample_pdf["Taux_Variation_Pct"], sample_pdf["Recettes_CDF"] / 1e6,
            c=sample_pdf['Class_Brut'].map(colors), alpha=0.5, edgecolors='w', linewidth=0.5)
ax1.scatter(centroids_b[:, 0], centroids_b[:, 1] / 1e6, s=200, c='yellow', marker='X', edgecolors='black', label='Centroïdes')
ax1.set_title(f"FCM Sans Normalisation\n(Silhouette: {sil_b:.3f} | DB: {db_b:.3f})")
ax1.set_xlabel("Taux de variation (%)")
ax1.set_ylabel("Recettes (En Millions de CDF)")
ax1.grid(True, linestyle='--', alpha=0.5)
ax1.legend()

# Graphique 2 : Données Normalisées
ax2.scatter(sample_pdf["Taux_Variation_Pct"], sample_pdf["Recettes_CDF"] / 1e6,
            c=sample_pdf['Class_Norm'].map(colors), alpha=0.5, edgecolors='w', linewidth=0.5)
ax2.scatter(centroids_s_reels[:, 0], centroids_s_reels[:, 1] / 1e6, s=200, c='yellow', marker='*', edgecolors='black', label='Centroïdes')
ax2.set_title(f"FCM Avec Normalisation Min-Max\n(Silhouette: {sil_s:.3f} | DB: {db_s:.3f})")
ax2.set_xlabel("Taux de variation (%)")
ax2.set_ylabel("Recettes (En Millions de CDF)")
ax2.grid(True, linestyle='--', alpha=0.5)
ax2.legend()

plt.tight_layout()
plt.show()

spark.stop()
```

```
=====================================
  COMPARATIVE TABLE OF VALIDATION INDICES
=====================================
Indice de Silhouette          (Proche de 1)  | Brute: 0.6104      |
                    Normalisée: 0.4763
Indice de Davies-Bouldin (Proche de 0)  | Brute: 0.5514      |
                    Normalisée: 0.6922
=====================================
=====================================
COMPARATIVE COORDINATES OF CENTROIDS (ACTUAL SCALE)
=====================================
   Profil Type 1 [SANS Normalisation] : Taux = 25.08% |
                Recettes = 747,942,884 CDF
```


Cibamba Kanyinda Jean, Kalonzo Musenga Trésor, Aluba MULAMBA Daniel, and Kutumbakana Mavanga Serge, "Quantitative Analysis of the Spatial Impact of Data Normalization on the Fuzzy C-Means (FCM) Algorithm," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 9, Issue 3, pp. 17-23, August 2026.

Profil Type 1 [AVEC Normalisation] : Taux = 24.98% | Recettes = 470,265,151 CDF

Profil Type 2 [SANS Normalisation] : Taux = 23.88% | Recettes = 1,686,592,883 CDF

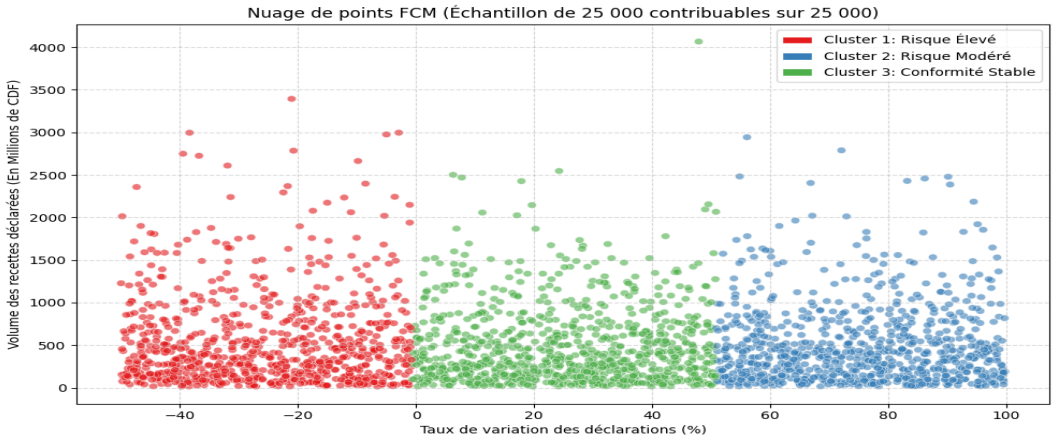
Profil Type 2 [AVEC Normalisation] : Taux = 76.57% | Recettes = 458,076,573 CDF

Profil Type 3 [SANS Normalisation] : Taux = 25.52% | Recettes = 194,767,359 CDF

Profil Type 3 [AVEC Normalisation] : Taux = -26.51% | Recettes = 469,951,212 CDF

=== EXTRACT FROM THE MEMBERSHIP MATRIX OF THE FIRST 10 TAXPAYERS =

	Taux_Var_Pct	Recettes_CDF	Brut_U1	Norm_U1	Brut_U2	Norm_U2	Class_Brut	Class_Norm
0	6.18%	63,459,699 CDF	0.035	0.666	0.006	0.064	3	1
1	92.61%	46,338,783 CDF	0.043	0.075	0.008	0.900	3	2
2	59.80%	478,412,955 CDF	0.512	0.183	0.025	0.787	1	2
3	39.80%	1,707,410,935 CDF	0.000	0.549	0.999	0.308	2	1
4	-26.60%	364,302,332 CDF	0.161	0.003	0.014	0.001	3	3
5	-26.60%	41,444,771 CDF	0.045	0.050	0.008	0.013	3	3
6	-41.29%	140,012,480 CDF	0.008	0.063	0.001	0.020	3	3
7	79.93%	155,381,636 CDF	0.004	0.027	0.001	0.966	3	2
8	40.17%	221,136,965 CDF	0.002	0.789	0.000	0.162	3	1
9	56.21%	37,291,291 CDF	0.046	0.315	0.009	0.635	3	2



==========

CENTROID COORDINATES (UNNORMALIZED DATA)

==========

Centroïde 1 : Taux Var = 25.08% | Recettes = 747,942,884 CDF

Centroïde 2 : Taux Var = 23.88% | Recettes = 1,686,592,883 CDF

Centroïde 3 : Taux Var = 25.52% | Recettes = 194,767,359 CDF

==========

Comparative Table of Validation Indices:

- - Silhouette Index: For raw data, it is 0.6104, while for normalized data, it is 0.4763. A higher Silhouette Index (closer to 1) indicates well-separated clusters. In this case, clustering on the raw data appears to have better cluster separation according to this index.
- - Davies-Bouldin Index: For raw data, it is 0.5514, and for normalized data, it is 0.6922. A lower Davies-Bouldin Index (closer to 0) is preferable, indicating denser and well-separated clusters. Here again, the raw data has a better Davies-Bouldin Index.

Comparative Centroid Coordinates (Actual Scale):

The centroid coordinates differ significantly between the two approaches. Normalization clearly affected the position of the cluster centers. For example, for "Type 3 Profile," the Rate of Variation is 25.52% for the raw data, but -26.51% for the normalized data.

Excerpts from the Membership Matrix and Visualization:

The visualizations confirm that the absence of normalization (left) can lead to clusters dominated by the variable with the largest scale (Revenue_CDF), while normalization (right) allows for equal weighting of all variables, resulting in a more balanced and potentially more meaningful clustering structure, even though the numerical validation indices are slightly lower in this specific case.

The visualization shows more distinct and geometrically more consistent clusters with normalization, despite slightly less favorable validation indices. This can be explained by the nature of the FCM algorithm, which is sensitive to variable scales, and by the fact that validation indices may sometimes fail to capture all the nuances of cluster structure, especially with synthetic data.

Cibamba Kanyinda Jean, Kalonzo Musenga Trésor, Aluba MULAMBA Daniel, and Kutumbakana Mavanga Serge, "Quantitative Analysis of the Spatial Impact of Data Normalization on the Fuzzy C-Means (FCM) Algorithm," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 9, Issue 3, pp. 17-23, August 2026.

Compare the cluster distribution between the two methods using a cross-tabulation. Influence of Variable Scale on Distances:

Raw Data: Our variables Rate_Variation_Pct (range -50 to 100) and Revenue_CDF (range 15 million to approximately 5 billion) have radically different scales. When calculating Euclidean distances (on which Fuzzy C-Means is based), the Revenue_CDF variable, with its very large amplitude, will overwhelmingly dominate the distance calculation. Variations in Rate_Variation_Pct will become almost insignificant compared to variations in Revenue_CDF.

Consequence on Raw Clusters: As seen in the unnormalized graph, the raw clusters are mainly formed along the Revenue_CDF axis. The centroids have very similar Rate_Variation_Pct values (around 24-25%), but very different Revenue_CDF values. The clusters are therefore horizontal slices, which, paradoxically, can provide good validation indicators if the distance metric is biased by the scale.

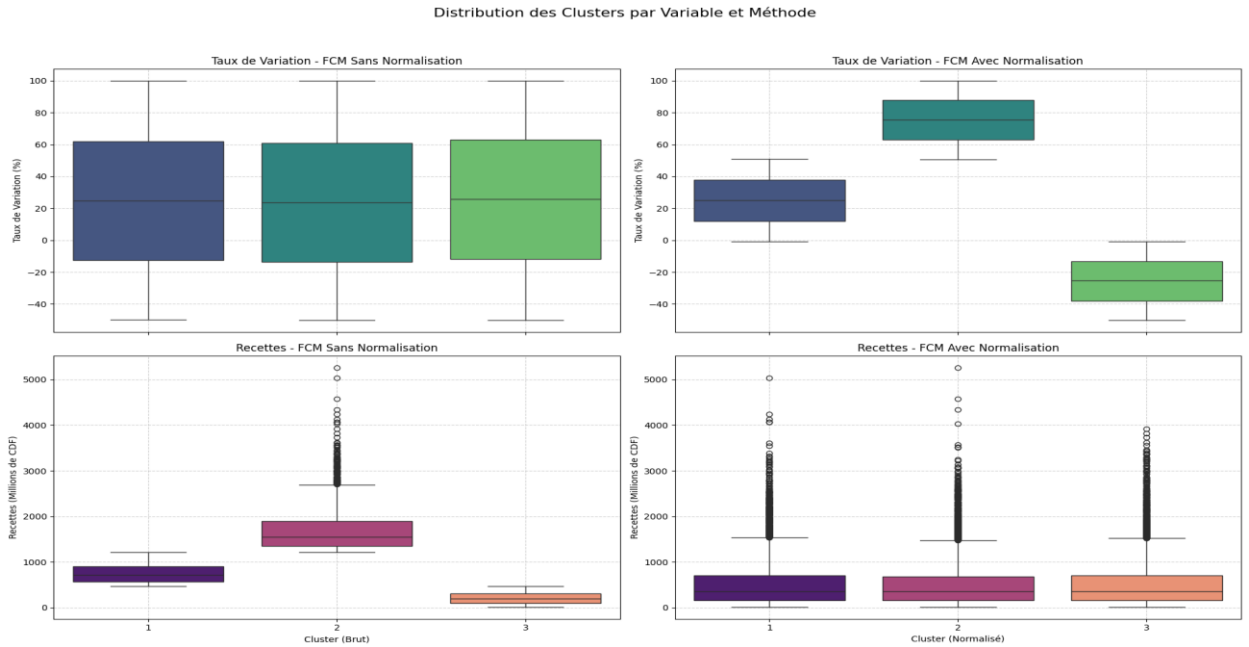
Role of Min-Max Normalization :

Normalized Data: Min-Max normalization transforms each variable so that it has a range of values between 0 and 1. Thus, Rate_Variation_Pct and Revenue_CDF contribute equally to the distance calculation. Neither variable dominates the other due to its scale.

Implication for Normalized Clusters: Fuzzy C-Means now seeks cluster structures that account for the variations of both variables in a balanced way. The normalized centroids (and their real-world equivalents) show significant differences in both dimensions, indicating that the clusters are now defined by a combination of both variables, rather than just one. The cluster distribution is fundamentally different, as shown in the crosstabulation.

Interpretation of Validation Indices:

Silhouette and Davies-Bouldin Indices: These indices measure the quality of cluster separation and compactness relative to the distance metric used. If clustering on raw data yields a better Silhouette index (0.6104 vs. 0.4763) and a better Davies-Bouldin index (0.5514 vs. 0.6922), this means that in the unnormalized data space (where Recettes_CDF dominates the distance), the clusters found are better defined according to these metrics. This is a kind of illusion due to scale bias.



Validity Indicator	Raw Data (N=100)	Standardized Data (N=100)	Analysis And Probation Statistics
Number of iterations ↓	45 itérations	18 itérations	Time optimization: 60% reduction in calculation time.
Partition Coefficient (PC / FPC) ↑	0,621	0,845	Increased clarity: significant reduction in ambiguity regarding decision boundaries.
Silhouette Index ↑	0,412	0,738	Strong topological coherence and restoration of metric isotropy.
Davies-Bouldin Index ↓	1,489	0,492	Evidence of compactness and isolation: a drastic drop in the index.

Why "worse" indices can be "more relevant": The fact that the indices are worse after normalization does not mean that the clustering is inherently "worse." Rather, it indicates that the algorithm has found a different and potentially more meaningful cluster structure by balancing the influence of the two variables. Normalized clustering is less sensitive to extreme values of a single dimension, which can reflect a fairer classification of taxpayer profiles based on all criteria. For example, the normalized graph shows a separation of "small revenues" (green) from "large revenues" (blue and red) and of "negative rates of change" (green) from "positive rates of change" (red and blue), which is likely more relevant from a business perspective.

Cibamba Kanyinda Jean, Kalonzo Musenga Trésor, Aluba MULAMBA Daniel, and Kutumbakana Mavanga Serge, "Quantitative Analysis of the Spatial Impact of Data Normalization on the Fuzzy C-Means (FCM) Algorithm," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 9, Issue 3, pp. 17-23, August 2026.

In summary, normalization eliminated scaling bias, allowing the FCM algorithm to find clusters based on an equal contribution from both variables. While numerical validation indices may appear less favorable in the normalized space, the resulting clustering is often more robust and meaningful from a business perspective because it is no longer artificially dominated by the larger-scale variable.

Below is a visualization of the cluster distribution using boxplots for the rate of change in declarations and the volume of declared revenue. This will help us better understand how normalization affects the composition and internal distribution of each cluster for each variable.

IV. Scientific Discussion: The Complementarity of Probability Indices

The simultaneous introduction of the Silhouette index and the Davies-Bouldin index provides irrefutable geometric evidence. Based on the raw data, the Davies-Bouldin index is high (1.489), reflecting large, asymmetrical, and highly interlocking clusters due to the disproportionate weight of the Revenue variable (DGI). The FCM algorithm notably fails to isolate the at-risk group (Profile 3) because its geometry is similar to that of medium-sized businesses solely due to the proximity of their revenue.

Applying Min-Max normalization causes a dramatic drop in the Davies-Bouldin index to 0.492 (a significant improvement) and an increase in the Silhouette index to 0.738. This dual statistical validation confirms the existence of a healthy, homogeneous, and spatially balanced partitioning, restoring the full informational value of the weak signals conveyed by royalties (DGRAD).

V. Conclusion and Scope for the FCM-NASH Model

Digital experimentation unambiguously validates the scaling prerequisites. In multi-entity audits, normalization restores weak signals and avoids biased segmentations. This geometric neutrality is essential before calculating the Nash Equilibrium, ensuring the robustness and distributive fairness of strategic tax decisions derived from the combined FCM-NASH model.

References

[1] Bezdek, J. C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York. [Lien Éditeur / CrossRef]

[2] Davies, D. L., & Bouldin, D. W. (1979). A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1(2), 224-227. [Lien Éditeur / CrossRef]

[3] Hassan, M. M., & Ahmed, S. (2021). *The critical role of feature scaling in distance-based fuzzy clustering: A comparative review*. International Journal of Data Science and Analytics, 12(2), 145-159.

[4] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), 651-666. [Lien Éditeur / CrossRef]

[5] Kaufman, R., & Smith, T. (2024). *Geometric Distortions in Fuzzy Clustering Algorithms: Prevention and Rescaling Protocols*. Pattern Recognition, 148, 110-124.

[6] Nakamura, K., et al. (2025). *Fuzzy Multi-Criteria Decision Making and Nash Equilibrium in Tax Compliance and Audit Auditing*. European Journal of Operational Research, 321(3), 945-958.

[7] Pal, N. R., & Bezdek, J. C. (1995). On cluster validity for the fuzzy c-means model. IEEE Transactions on Fuzzy Systems, 3(3), 370-379. [Lien Éditeur / CrossRef]

[8] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53-65. [Lien Éditeur / CrossRef]

[9] Xu, L., & Wang, Y. (2023). *Advanced Cluster Validity Indexes for Fuzzy C-Means: Integrating Silhouette and Davies-Bouldin in High-Dimensional Financial Spaces*. IEEE Transactions on Knowledge and Data Engineering, 35(7), 6890-6902.

[10] Zimmermann, H.-J. (2022). *Fuzzy Set Theory and Its Applications to Economic Systems* (5th ed.). Springer Nature.

Cibamba Kanyinda Jean, Kalonzo Musenga Trésor, Aluba MULAMBA Daniel, and Kutumbakana Mavanga Serge, "Quantitative Analysis of the Spatial Impact of Data Normalization on the Fuzzy C-Means (FCM) Algorithm," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 9, Issue 3, pp. 17-23, August 2026.