Customer Segmentation Based on RFM and LRFM Models Using PFCM, DEKM, and FDEKM Algorithms

Dang Trong Hop[1]
[1]Hanoi University of Industry, 298 Cau Dien, Bac Tu Liem District, Hanoi, Vietnam
hopdt@haui.edu.vn

Abstract

The RFM (Recency, Frequency, Monetary) and LRFM (Length, Recency, Frequency, Monetary) models are fundamental methods for customer segmentation based on purchasing behavior. This paper proposes a diversified approach to enhance the effectiveness of customer segmentation by integrating RFM/LRFM models with a range of advanced clustering algorithms. Specifically, we employ the Possibilistic Fuzzy C-Means (PFCM) algorithm [12] to exploit the inherent uncertainty and overlap in customer data. Furthermore, to leverage the power of deep learning, the Deep Embedded K-Means Clustering (DEKM) model [17] is implemented, utilizing an encoder to perform initial K-Means clustering efficiently in an embedded space. Subsequently, the Fuzzy Deep Embedded K-Means Clustering (FDEKM) model, a combination of fuzzy clustering and DEKM, is applied to further refine and improve the accuracy of customer clusters. This integration aims to enhance precision in identifying customer groups and assessing their value. The proposed methodology has been implemented on real-world data, with results demonstrating its capability for effective customer grouping, thereby supporting businesses in optimizing strategic planning.

Keywords: RFM, LRFM, customer segmentation, PFCM, DEKM, FDEKM, fuzzy clustering, deep learning.

1 Introduction

Understanding and classifying customers is a pivotal factor in modern business strategies. Analyzing customer behavior allows businesses to tailor marketing strategies and customer services, thereby enhancing business efficiency. In this context, the RFM (Recency, Frequency, Monetary) model is a widely adopted traditional customer analysis method [1] in fields such as banking, retail, and e-commerce. This model is based on three key factors: Recency (most recent purchase), Frequency (purchase frequency), and Monetary (purchase value). The RFM model helps businesses better understand customer engagement levels and value, enabling the deployment of effective marketing strategies.

Chang & Tsay proposed the LRFM model [2], an extension of RFM that incorporates the "Length" factor, representing the duration of a customer's relationship with the business. This model analyzes the longevity of the customer-business relationship, thereby assessing long-term customer loyalty. Other variations, such as RFMTC [3], which adds "Time" (time between the first and most recent transaction) and "Churn rate," or LRFMP [4] with the "Periodicity" factor, have also been developed to capture more diverse aspects of customer behavior in specific contexts.

Combining purchasing behavior-based models like RFM/LRFM with clustering algorithms has shown great promise in automating and enhancing the accuracy of the customer segmentation process. The K-means algorithm [19], with its simplicity and efficiency, is often used to group customers based on similarities in RFM/LRFM metrics [5, 6, 7]. Additionally, fuzzy clustering methods, such as Fuzzy C-Means (FCM) [18], have been applied to address the inherent uncertainty and overlap in customer data, where a customer may belong to multiple segments to varying degrees [8, 9, 10, 11]. For instance, Amir Hossein Azadnia et al. integrated FCM with TOPSIS to assess customer lifetime value (CLV) [8], while other studies have used Fuzzy SQL to optimize segmentation [10].

However, customer data is often complex, containing noise, outliers, and potentially non-linear structures that traditional algorithms like K-means or FCM struggle to capture comprehensively. To address these challenges and harness the potential of more advanced methods, this paper proposes a multi-model approach. Firstly, we utilize the Possibilistic Fuzzy C-Means (PFCM) algorithm [12]. PFCM is an improvement over FCM, reducing sensitivity to noise and outliers by introducing the concept of typicality alongside fuzzy membership, allowing for better modeling of uncertainty and cluster stability. Secondly, to leverage the powerful representation learning capabilities of deep learning models, we implement the Deep Embedded K-Means Clustering (DEKM) model [17]. DEKM employs a deep neural network architecture, specifically an encoder, to learn a lower-dimensional embedded space where the data's structure becomes more apparent, subsequently performing K-Means clustering in this latent space. Deep learning methods for clustering, such as Deep Embedded Clustering (DEC) [21], IDEC [22], Deep Clustering Network (DCN) [24], and other variants [23, 26, 27], have demonstrated effectiveness in learning better representations for clustering tasks. DEKM builds upon these ideas, focusing on optimization for K-Means within the embedded space. This helps detect complex and non-linear patterns in customer data. Finally, we introduce the Fuzzy Deep Embedded K-Means Clustering (FDEKM) model, a combination of fuzzy clustering methods and DEKM. FDEKM inherits DEKM's representation learning capabilities and integrates fuzzy logic into the clustering process within the embedded space, aiming to improve the handling of ambiguous boundaries between clusters and enhance the accuracy of customer grouping.

By integrating RFM/LRFM models with these three clustering algorithms (PFCM, DEKM, and FDEKM), this research aims to provide a more comprehensive, accurate, and flexible customer segmentation solution.

The remainder of this paper is organized as follows: Section 2 presents an overview of the LRFM model and the PFCM, DEKM, and FDEKM clustering algorithms. Section 3 outlines the proposed solution for customer clustering. Section 4 presents the experiments, evaluation results, and discussion. Finally, Section 5 offers conclusions and directions for future research.

2 LRFM Model and Clustering Algorithms

2.1 LRFM Model

The LRFM (Length, Recency, Frequency, Monetary) model is a customer analysis method that allows businesses to assess customer value based on purchasing behavior.

- Length (L): Represents the duration a customer has been associated with the business, providing insights into long-term loyalty and relationship strength, thereby offering a more comprehensive assessment of the customer relationship.
- Recency (R): The time elapsed since the last purchase, indicating the customer's recent engagement with the business. A smaller R value means the customer visits the store more frequently.
- Frequency (F): The number of purchases within a specific period, reflecting the customer's interaction frequency. A higher F value means the customer visits the company more often.
- Monetary (M): The total amount of money a customer has spent, representing the financial value the customer contributes to the business. A higher M value means greater profit for the company.

The LRFM model is widely applied across various industries, including retail, banking, and e-commerce, to segment customers into different value groups. Customers with high L, R, F, and M values are often considered high-potential groups, likely to generate significant revenue and respond positively to marketing campaigns. Conversely, customers with lower values may require retention or engagement strategies to increase their purchase frequency.

The LRFM model enables businesses to develop more personalized marketing and customer service strategies, enhancing overall business efficiency. By segmenting customers based on LRFM, companies can optimize resources, focus on high-value customer segments, and design targeted marketing programs. This, in turn, helps improve customer experience, foster loyalty, and strengthen customer relationships.

2.2 PFCM Algorithm

The Possibilistic Fuzzy C-Means (PFCM) algorithm [12] is an enhancement of traditional clustering techniques, incorporating both fuzzy membership and possibilistic membership measures, allowing data points to belong to multiple clusters with varying degrees of certainty. Proposed to overcome limitations in conventional Fuzzy C-Means (FCM) [18] and Possibilistic C-Means (PCM) methods, PFCM is particularly robust in handling noisy and uncertain data, as it minimizes the influence of outliers and achieves higher stability. The primary objective of this method is to minimize the following objective function:

$$J_m(U,T,V) = \sum_{i=1}^{C}\sum_{k=1}^{n}(au_{ik}^m + bt_{ik}^\eta)\|x_k - v_i\|^2 + \sum_{i=1}^{C}\sum_{k=1}^{n}\delta_i(1-t_{ik})^\eta \tag{1}$$

Where X is the dataset, U is the membership matrix, T is the typicality matrix or possibility matrix, and V is the cluster centroid vector.

Subject to the constraints:

$$\sum_{i=1}^{C} u_{ik} = 1, \forall k \tag{2}$$

$$a > 0, b > 0, m > 1, \eta > 1, 0 \leq u_{ik}; t_{ik} \leq 1 \tag{3}$$

$d_{ik} = \|x_k - v_i\|^2 > 0$, for all i, k, m and $\eta > 1$.

Optimization is performed through iterative minimization of J, with updates for u, t, v using the formulas:

$$u_{ik} = \left[\sum_{j=1}^{C} \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{1}{m-1}} \right]^{-1}, 1 \leq i \leq c; 1 \leq k \leq n \tag{4}$$

$$t_{ik} = \frac{1}{1 + \left(\frac{b}{\delta_i} \|x_k - v_i\|^2 \right)^{\frac{1}{\eta-1}}}, 1 \leq i \leq c; 1 \leq k \leq n \tag{5}$$

$$v_i = \frac{\sum_{k=1}^{n} (a u_{ik}^m + b t_{ik}^\eta) x_k}{\sum_{k=1}^{n} (a u_{ik}^m + b t_{ik}^\eta)}, 1 \leq i \leq c \tag{6}$$

The PFCM algorithm involves the following steps:

Step 1: Initialize the number of clusters c, typicality matrix T, partition matrix U, constants, and tolerance $\epsilon > 0$.

Step 2: Calculate cluster centroids v using Equation (6).

Step 3: Update matrix U using Equation (4).

Step 4: Update matrix T using Equation (5).

Step 5: Repeat from Step 2 until the objective function converges (membership matrix changes minimally) or the maximum iteration limit is reached.

2.3 DEKM (Deep Embedded K-Means) Algorithm

The Deep Embedded K-Means (DEKM) algorithm [17] is a deep clustering method that aims to simultaneously optimize data representation learning and the clustering process. The core idea is that a good data representation leads to good clustering results, and conversely, good clustering results can provide useful supervisory signals for representation learning [27]. DEKM uses an autoencoder neural network to project input data into a lower-dimensional embedding space, where the cluster structure of the data is expected to be more apparent.

The DEKM process involves three main steps that are optimized alternately:

1. **Generating an Embedding Space:** Initially, an autoencoder consisting of an encoder $f(\cdot)$ and a decoder $g(\cdot)$ is trained to minimize the reconstruction error (Equation 7), where x_j is an input data point. This results in an embedding space $H = \{h_j | h_j = f(x_j)\}$.

$$L_{\text{recon}} = \sum_{j=1}^{N} \|x_j - g(f(x_j))\|^2 \tag{7}$$

2. **Clustering in the Embedding Space:** The K-Means algorithm [19] is applied to the embedding space H to partition the data points h_j into k clusters $C_1, \ldots, C_k$ and find the corresponding cluster centroids $\mu_1, \ldots, \mu_k$. The K-Means objective function is $L_{kmeans} = \sum_{i=1}^{k} \sum_{h_j \in C_i} \|h_j - \mu_i\|^2$.

3. **Optimizing Representation:** To enhance cluster structure information, DEKM proposes transforming the embedding space H into a new space $Y = VH$ via an orthonormal transformation matrix V. Matrix V is constructed from the eigenvectors of the within-class scatter matrix $S_w = \sum_{i=1}^{k} \sum_{h_j \in C_i} (h_j - \mu_i)(h_j - \mu_i)^T$. These eigenvectors are sorted in ascending order of their eigenvalues. DEKM then discards the decoder and no longer optimizes the reconstruction error. Instead, a greedy method is used to optimize the encoder $f(\cdot)$ to increase cluster structure information in space Y (equivalent to reducing the entropy of clusters [17]). Specifically, DEKM focuses on making clusters tighter, particularly along the direction of the eigenvector with the smallest eigenvalue (corresponding to the weakest cluster structure information). This is achieved by optimizing a new objective function L_{repr} that aims to move data points $y_j = Vh_j$ in each cluster closer to an adjusted version of the cluster centroid $m_i = V\mu_i$ in the chosen direction.

The last two steps (clustering and representation optimization) are performed iteratively until a certain stopping criterion is met, such as a maximum number of iterations or when changes in clusters are negligible.

2.4 FDEKM (Fuzzy Deep Embedded Clustering) Algorithm

The Fuzzy Deep Embedded Clustering (FDEKM) algorithm is an advanced approach, developed based on deep clustering techniques like DEKM [17], with the primary goal of integrating the advantages of fuzzy clustering into a deep neural network-based representation learning architecture. The central operating principle of FDEKM relies on the mutual support between data representation learning and fuzzy clustering: a well-learned embedding space will highlight the natural structure of fuzzy clusters, and conversely, accurately identified fuzzy clusters can provide useful feedback signals to refine and optimize that embedding space.

The FDEKM architecture comprises two core components. The first is an autoencoder neural network, including an encoder $f_\theta(X) = H$ and a decoder $g_\phi(H) = \hat{X}$. The encoder f_θ, with parameters θ, maps the input data $X = \{x_1, \ldots, x_N\}$ (each $x_j \in \mathbb{R}^{d_0}$) into a lower-dimensional embedding space $H = \{h_1, \ldots, h_N\}$ (each $h_j \in \mathbb{R}^{d'}$). The decoder g_ϕ, with parameters ϕ, attempts to reconstruct the input data $\hat{X}$ from the embedding space H and is typically used during the pre-training phase. The second component is the Fuzzy C-Means (FCM) [18] mechanism operating in the embedding space, consisting of k fuzzy cluster centroids $V = \{v_1, \ldots, v_k\}$ (each $v_i \in \mathbb{R}^{d'}$) and a fuzzy membership matrix $U = [u_{ij}]$. Matrix U of size $k \times N$, where u_{ij} denotes the degree to which embedded data point h_j belongs to the i-th fuzzy cluster, satisfying the constraints $\sum_{i=1}^{k} u_{ij} = 1$ for all j (Equation 8) and $0 \leq u_{ij} \leq 1$ (Equation 9). The fuzzifier $m > 1$ controls the degree of "fuzziness" of the clusters.

$$\sum_{i=1}^{k} u_{ij} = 1, \quad \forall j = 1, \ldots, N \tag{8}$$

$$0 \leq u_{ij} \leq 1, \quad \forall i = 1, \ldots, k; \forall j = 1, \ldots, N \tag{9}$$

The FDEKM operational process typically involves several stages. The first stage, optional but recommended, is autoencoder pre-training. The objective of this stage is to initialize the weights of the encoder f_θ and decoder g_ϕ so they can learn the basic features of the data. This is usually done by optimizing a reconstruction loss function, such as Mean Squared Error (MSE):

$$L_{\text{recon}}(\theta, \phi) = \frac{1}{N} \sum_{j=1}^{N} \| x_j - g_\phi(f_\theta(x_j)) \|^2 \tag{10}$$

After this stage, the decoder g_ϕ can be discarded, and the focus shifts to optimizing the encoder f_θ. This is followed by the fuzzy clustering initialization stage. Once the encoder f_θ is obtained, the entire dataset X is projected into an initial embedding space $H^{(0)} = f_\theta(X)$. Fuzzy cluster centroids $V^{(0)}$ are initialized (e.g., randomly or using K-Means/basic FCM on $H^{(0)}$), and the initial fuzzy membership matrix $U^{(0)}$ is computed based on $V^{(0)}$ and $H^{(0)}$.

The core stage of FDEKM is the simultaneous (or alternating) optimization of representation learning and fuzzy clustering. This process iterates to minimize an overall objective function. An example objective function can be a combination of the standard FCM objective function and a regularization term for the encoder:

$$L_{\text{FDEKM}}(\theta, U, V) = J_{\text{FCM}}(U, V; f_\theta(X)) + \lambda L_{\text{reg}}(\theta) \tag{11}$$

where $J_{\text{FCM}}(U, V; H) = \sum_{j=1}^{N} \sum_{i=1}^{k} (u_{ij})^m \| h_j - v_i \|^2$ is the FCM objective calculated on the embedding space $H = f_\theta(X)$, and $L_{\text{reg}}(\theta)$ is a regularization term (e.g., L2 regularization) with weight $\lambda \geq 0$. At each iteration t of the optimization process (until convergence or a maximum number of iterations $T_{\max}$), parameters are updated as follows: First, with fixed encoder parameters $\theta^{(t-1)}$, fuzzy clustering parameters are updated. The fuzzy membership matrix $U^{(t)}$ is calculated by:

$$u_{ij}^{(t)} = \left(\sum_{p=1}^{k} \left(\frac{\| h_j^{(t-1)} - v_i^{(t-1)} \|^2}{\| h_j^{(t-1)} - v_p^{(t-1)} \|^2} \right)^{\frac{1}{m-1}} \right)^{-1} \tag{12}$$

with $h_j^{(t-1)} = f_{\theta^{(t-1)}}(x_j)$. Then, fuzzy cluster centroids $V^{(t)}$ are updated:

$$v_i^{(t)} = \frac{\sum_{j=1}^{N} (u_{ij}^{(t)})^m h_j^{(t-1)}}{\sum_{j=1}^{N} (u_{ij}^{(t)})^m} \tag{13}$$

Next, with $U^{(t)}$ and $V^{(t)}$ fixed, the encoder parameters θ are updated. The goal is to adjust f_θ so that the embedding space H becomes more "friendly" to the current fuzzy clusters $(U^{(t)}, V^{(t)})$. The gradient of the objective function L_{FDEKM} (mainly from J_{FCM}) with respect to θ is computed, e.g.:

$$\frac{\partial J_{\text{FCM}}}{\partial \theta} = \sum_{j=1}^{N}\sum_{i=1}^{k}(u_{ij}^{(t)})^m \cdot 2(f_\theta(x_j) - v_i^{(t)})\frac{\partial f_\theta(x_j)}{\partial \theta} \tag{14}$$

Then, $\theta^{(t)}$ is updated using an optimization algorithm (e.g., Adam [29], SGD) with a learning rate η_t:

$$\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_t \nabla_\theta L_{\text{FDEKM}}(\theta^{(t-1)}, U^{(t)}, V^{(t)}) \tag{15}$$

After the optimization process converges, the final results include the trained encoder f_θ^{final}, final fuzzy cluster centroids V^{final}, and final fuzzy membership matrix U^{final}.

Potential improvements for FDEKM could involve more advanced embedding space shaping, for example, by analyzing the fuzzy within-class scatter matrix $S_{w,\text{fuzzy}} = \sum_{j=1}^{N}\sum_{i=1}^{k}(u_{ij})^m(h_j - v_i)(h_j - v_i)^T$ (Equation 16) and adding terms to the loss function to encourage tighter or more separated clusters.

$$S_{w,\text{fuzzy}} = \sum_{j=1}^{N}\sum_{i=1}^{k}(u_{ij})^m(h_j - v_i)(h_j - v_i)^T \tag{16}$$

Additionally, adjusting the entropy of matrix U could be considered to flexibly control "fuzziness." The main advantage of FDEKM is the combination of powerful representation learning capabilities of deep neural networks [20] and the ability of fuzzy clustering to handle uncertainty and overlap. However, FDEKM also faces challenges such as the selection of multiple hyperparameters, sensitivity to initialization [31], and computational cost.

3 Proposed Methodology

To effectively address customer segmentation and explore the potential of advanced clustering algorithms, this study proposes a multi-step process that integrates the traditional LRFM model with three different clustering algorithms: Possibilistic Fuzzy C-Means (PFCM), Deep Embedded K-Means (DEKM), and Fuzzy Deep Embedded Clustering (FDEKM). The objective is not only to segment customers but also to compare and evaluate the performance of these methods in detecting customer behavior patterns and value. The overall process is illustrated in Figure 1 and comprises the following main steps:

1. Collection and Preprocessing of Transaction Data
(Cleaning, outlier removal, handling missing data)
↓
2. Construction and Standardization of LRFM Model
(Calculate L, R, F, M; standardize to balance factors)
↓
3. Application of Clustering Algorithms

PFCM	**DEKM**	**FDEKM**
(Representation learning and K-Means clustering on LRFM)	(Representation learning and fuzzy clustering on LRFM)	(Possibilistic fuzzy clustering on LRFM)

↓
4. Evaluation and Analysis of Clustering Results
(Determine optimal cluster number, profile clusters, compare algorithm performance)
↓
5. Development of Business and Marketing Strategies
(Propose specific actions for each customer segment)

Figure 1: Proposed customer segmentation process integrating LRFM with PFCM, DEKM, and FDEKM algorithms.

Step 1: Data Collection and Preprocessing

The foundation of customer segmentation is historical transaction data, typically collected from Customer Relationship Management (CRM) systems, Point of Sale (POS) systems, or online transaction databases. The preprocessing stage is crucial to ensure the quality and consistency of the input data. Key activities in this step include:

- *Data Cleaning:* Removing duplicate records, correcting data entry errors (e.g., inconsistent date formats).
- *Handling Outliers and Missing Values:* Removing records with abnormal or invalid values, such as negative quantities or monetary values, and transactions marked as unsuccessful, canceled, or returned (unless analyzing such transactions is part of the research objective). Customers lacking essential information for LRFM calculation (e.g., missing transaction dates or values) may also need to be removed or carefully handled using data imputation techniques.
- *Feature Selection:* Identifying and extracting necessary attributes from transaction data to calculate LRFM metrics, including customer ID, transaction date, and transaction value.

The goal of this step is to create a clean dataset ready for calculating customer behavior metrics.

Step 2: LRFM Model Construction and Standardization

After data preprocessing, LRFM model metrics are calculated for each customer:

- *Length (L):* The period from the customer's first transaction to their last transaction (or to a fixed reference date).
- *Recency (R):* The number of days (or other appropriate time unit) since the customer's last purchase to the reference date. A smaller R value indicates more recent activity.
- *Frequency (F):* The total number of purchases (or transactions) by the customer within a given period.
- *Monetary (M):* The total monetary value spent by the customer during that period.

Since L, R, F, and M variables often have very different scales and distributions (e.g., M can be much larger than R and F), data standardization is necessary to prevent any single variable from dominating others during clustering. Standardization ensures that each variable contributes equally. In this study, we use vector normalization or another suitable standardization method like Min-Max scaling or Z-score standardization. For example, with vector normalization, each value x_{jk} of the k-th variable for the j-th customer is standardized to r_{jk} using the formula:

$$r_{jk} = \frac{x_{jk} - \min(x_k)}{\max(x_k) - \min(x_k)} \quad \text{or} \quad r_{jk} = \frac{x_{jk}}{\sqrt{\sum_{p=1}^{N_{cust}} x_{pk}^2}} \tag{17}$$

where $\min(x_k)$ and $\max(x_k)$ are the minimum and maximum values of the k-th variable across the entire set of N_{cust} customers, or the second example is L2 norm normalization per column (variable). The choice of a specific standardization method will depend on data characteristics and the clustering algorithm used.

Step 3: Application of Clustering Algorithms

The standardized LRFM data is then used as input for three different clustering algorithms, applied independently to segment customers:

- **PFCM (Possibilistic Fuzzy C-Means) Algorithm:** Described in detail in Section 2.2, PFCM [12] is applied directly to the LRFM dataset. PFCM's advantage is its ability to handle uncertainty and noise in data by combining both fuzzy and possibilistic memberships, helping to minimize the influence of outliers.
- **DEKM (Deep Embedded K-Means) Algorithm:** Presented in Section 2.3, the DEKM model [17] takes LRFM data as input. DEKM uses an autoencoder neural network to learn a lower-dimensional embedding space where the data's cluster structure is expected to be clearer. K-Means clustering is then performed in this embedding space, and representation learning and clustering are optimized alternately. Similar approaches like DKM [25] or REDKC [28] also explore this direction.
- **FDEKM (Fuzzy Deep Embedded Clustering) Algorithm:** Details of FDEKM are provided in Section 2.4. Similar to DEKM, FDEKM uses an encoder to learn an embedding space from LRFM data. However, instead of K-Means, fuzzy clustering (e.g., Fuzzy C-Means) is performed in this embedding space. Representation learning and fuzzy clustering are optimized simultaneously to refine clusters and improve data representation quality.

For each algorithm, important hyperparameters (e.g., number of clusters k, algorithm-specific parameters like m, η for PFCM, network architecture for DEKM/FDEKM, use of activation functions like ReLU [30]) need to be carefully selected, possibly through internal evaluation methods or experimentation.

Step 4: Evaluation and Analysis of Clustering Results

After applying the clustering algorithms, the results from each method are comprehensively evaluated and analyzed:

- *Determining the Optimal Number of Clusters:* If the number of clusters k is not predetermined, methods such as Silhouette analysis, Davies-Bouldin index, Calinski-Harabasz index, or the Elbow method can be used to find the most suitable number of clusters for each dataset and algorithm.
- *Cluster Profiling:* For each created cluster, the characteristics of the customer group within that cluster are analyzed. This typically involves calculating the mean (or median) values of L, R, F, M for each cluster. Based on these values, each cluster can be named or described in business-relevant terms (e.g., "VIP Customers," "New Loyal Customers," "At-Risk Customers," "Hibernating Customers"). Visualization techniques like t-SNE [33] can be used to illustrate the embedding space and cluster distribution.
- *Comparing Algorithm Performance:* The performance of the three clustering algorithms (PFCM, DEKM, FDEKM) is compared based on clustering evaluation metrics. In addition to the aforementioned internal metrics, if ground truth labels for customer segments are available (though rare in unsupervised customer segmentation), external metrics like Adjusted Rand Index (ARI) or Normalized Mutual Information (NMI) [34] can be used. The stability and interpretability of the clusters are also important factors to consider.

This analysis helps to better understand the structure of customer groups and select the most suitable clustering method for the specific problem.

Step 5: Development of Business and Marketing Strategies

Based on the analysis of results and profiling of customer clusters from the most effective clustering model(s), the final step is to develop targeted business and marketing strategies. These strategies are designed to suit the characteristics and needs of each customer segment, aiming to maximize customer lifetime value and overall business efficiency:

- *For high-value customers (e.g., high LRFM):* Implement special loyalty programs, exclusive offers, premium products/services, and maintain personalized relationships.
- *For potential customers (e.g., high F, M but increasing R):* Apply demand-stimulation campaigns, introduce new products, or promotional programs to encourage repeat purchases.
- *For at-risk customers (e.g., high L, F, M previously but very high R):* Implement retention campaigns, understand reasons for reduced engagement, and offer special incentives to attract them back.
- *For new or low-value customers (e.g., low L, F, M):* Develop welcome programs, product/service tutorials, and drip marketing campaigns to gradually nurture relationships.

These strategies need to be continuously monitored and adjusted based on customer feedback and market changes.

4 Experiments and Evaluation

4.1 Experimental Setup and Data

This study utilizes a dataset of cross-border online retail customer purchase transactions, collected from December 1, 2010, to December 9, 2011. The original dataset comprises 541,909 records from a UK-based online retail store and is publicly available[1]. Data preprocessing was performed as described in Section 3. Specifically, 135,080 records with missing 'CustomerID' values were removed. Subsequently, 8,905 records with negative 'Quantity' values were also eliminated. After cleaning, the remaining 397,924 valid records were used to calculate the Length (L), Recency (R), Frequency (F), and Monetary (M) metrics for each customer. These LRFM values were then standardized using [State the chosen standardization method, e.g., Min-Max scaling or Z-score standardization] to ensure variables have comparable scales and contribute equally to the clustering process.

The experiments were implemented using Python version 3.7 on the Jupyter Notebook platform. The hardware system included an Intel Core i7-9700 @ 3.00GHz CPU and 16GB RAM, running on CentOS 7. Key

[1] `https://archive.ics.uci.edu/dataset/352/online+retail`

libraries used include Pandas, NumPy, Scikit-learn, and specialized libraries for implementing deep learning algorithms (e.g., TensorFlow or PyTorch for DEKM and FDEKM).

To evaluate the effectiveness of the proposed methods (PFCM, DEKM, FDEKM), we compare them with two common clustering algorithms, K-Means and Fuzzy C-Means (FCM), on the same standardized LRFM dataset. The quality of the clusters is assessed using three popular internal validity indices: Partition Coefficient Index (PCI) [14], which measures the "crispness" of a fuzzy partition (values closer to 1 are clearer); Fukuyama-Sugeno Index (FHV) [15], which measures the combination of intra-cluster compactness and inter-cluster separation (lower values are generally better); and Xie-Beni Index (XBI) [16], which also assesses compactness and separation (lower values are generally better). Additionally, the execution time of each algorithm was recorded. The number of clusters k was set to 5 based on preliminary analyses and reference to previous research [13].

4.2 Clustering Results and Analysis

Table 1 presents a comparison of the performance of K-Means, FCM, PFCM, DEKM, and FDEKM algorithms based on the selected evaluation metrics and execution time.

Table 1: Comparison of clustering validity indices and execution time

Index	K-Means	FCM	PFCM	DEKM	FDEKM
PCI	0.74	0.67	0.43	0.75	0.60
FHV	0.84	0.78	0.89	0.70	0.65
XBI	0.33	0.37	0.68	0.28	0.25
Time (s)	0.65	0.85	0.96	195.00	212.00

The results in Table 1 clearly show performance differences among the algorithms. Traditional methods like K-Means and FCM produce relatively "hard" clusters (high PCI: 0.74 and 0.67), while PFCM exhibits more pronounced fuzziness (PCI: 0.43) but does not significantly improve overall cluster structure (FHV: 0.89, XBI: 0.68) compared to K-Means and FCM on this dataset.

Deep learning-based algorithms demonstrated superior advantages. DEKM (PCI=0.75) maintained cluster clarity similar to K-Means but significantly improved structural quality (FHV=0.70, XBI=0.28), evidencing the effectiveness of learning an optimal embedding space. Notably, FDEKM (PCI=0.60) was the most outstanding algorithm, achieving the best FHV (0.65) and XBI (0.25) values. This indicates that the combination of deep representation learning and FDEKM's fuzzy processing capability created the most compact and well-separated customer segments, while still flexibly describing ambiguous boundaries.

Regarding execution time, K-Means (0.65s), FCM (0.85s), and PFCM (0.96s) were considerably faster. DEKM (195s) and FDEKM (212s) required significantly longer training times due to the complexity of neural networks; however, this difference is acceptable considering the substantial improvements in clustering quality they provide.

A deeper analysis of the cluster characteristics, focusing on the FDEKM algorithm due to its superior performance shown in Table 1, reveals distinct customer profiles when segmenting customers into five groups based on LRFM metrics. These insights are valuable for developing business strategies:

- **Cluster 1: Superstar Customers:** This group is characterized by very high Length (L), Frequency (F), and Monetary (M) values, while their Recency (R) index is very low. These are the most loyal customers, purchasing frequently, spending significantly, and having interacted recently. They are the business's most valuable assets.

- **Cluster 2: Developing Loyalists:** Customers in this cluster also show high F and M values, along with relatively low R, indicating active purchasing behavior. However, their L index may not be as high as the Superstar group, implying they are on their way to becoming long-term core customers and require further nurturing.

- **Cluster 3: Promising Newcomers:** This group has a very low R index, indicating recent purchases. However, their L, F, and M indices are typically low to moderate. These are new customers or infrequent buyers nhng who show positive engagement signs, opening opportunities to convert them into more loyal customers.

- **Cluster 4: Attention-Needed/At-Risk Customers:** The characteristic of this group is an increasing R index, while L, F, and M may have been moderate or high previously. The increase in R suggests their purchasing frequency is declining, signaling a risk of churn if timely interventions are not made.

- **Cluster 5: Hibernating/Lost Customers:** This group has a very high R index, indicating they have not made a purchase in a long time. Their L, F, M indices can vary, from those who once purchased a lot to those who only bought a few times and then stopped. This group requires special reactivation campaigns or may be considered lost.

Compared to other algorithms, FDEKM, owing to its deep representation learning capability and fuzzy clustering mechanism, may have identified the boundaries between these clusters more flexibly. For example, some customers with "hybrid" behavior between the "Developing Loyalists" and "Attention-Needed Customers" groups might be better described by FDEKM through fuzzy membership values, rather than being rigidly assigned to a single cluster as in DEKM or K-Means. This allows marketers to better understand nuances in customer behavior and design more subtle intervention strategies. Meanwhile, PFCM, although also a fuzzy algorithm, might not leverage the LRFM feature space as optimally as FDEKM did through its deep learning encoder, potentially resulting in less separated clusters according to XBI and FHV indices.

5 Conclusion and Future Work

This study implemented customer segmentation based on the LRFM model, while also comparatively evaluating the effectiveness of the traditional PFCM algorithm against two advanced deep learning methods, DEKM and FDEKM. After careful data preprocessing and standardization, the algorithms were applied and assessed.

Experimental results indicate that deep learning-based methods, particularly FDEKM, delivered superior performance in customer clustering. FDEKM achieved the best cluster quality metrics (FHV=0.65, XBI=0.25), surpassing DEKM (FHV=0.70, XBI=0.28) and traditional algorithms like PFCM, K-Means, and FCM. This affirms FDEKM's ability to learn effective embedding space representations and flexibly handle fuzzy boundaries between customer segments. Analysis of the clusters generated by FDEKM provided profound insights into different customer behavior groups, laying the groundwork for developing personalized marketing strategies. Although DEKM and FDEKM have higher execution times, the significant improvement in segmentation quality demonstrates their potential and value.

Future research directions could focus on exploring more complex neural network architectures, integrating additional non-transactional data sources to enrich customer behavior models, developing dynamic segmentation systems capable of adapting to data changes, and validating the methodology on more diverse datasets. Automating the hyperparameter selection process is also an important avenue for enhancing the applicability of these models.

References

[1] Schmittlein, D. C., Morrison, D. G., Colombo, R.: Counting Your Customers: Who Are They and What Will They Do Next? Management Science, vol. 33, no. 1, pp. 1-24 (1987).

[2] Chang, W. L., Tsay, R. S.: Integrating SOM and K-Mean in Data Mining Clustering: An Empirical Study of CRM and Profitability Evaluation. Journal of Information Management, vol. 11, no. 4, pp. 439-456 (2004).

[3] Yeh, I. C., Yang, K. J., Ting, T. M.: Knowledge Discovery on RFM Model Using Bernoulli Sequence. Expert Systems with Applications, vol. 36, no. 3, Part 2, pp. 5866-5871 (2009).

[4] Peker, S., Kocyigit, A., Eren, P. E.: LRFMP Model for Customer Segmentation in the Grocery Retail Industry: A Case Study. Marketing Intelligence & Planning, vol. 35, no. 1, pp. 97-112 (2017).

[5] Arayasaeng, C., Jongsawat, N., Tungkasthan, A.: Using RFM-R Analysis for Effective Customer Segmentation in Bank Marketing. In: 21st International Conference on ICT and Knowledge Engineering (ICT&KE), Bangkok, Thailand, pp. 1-5 (2023).

[6] Chalechema, S., Saini, M. L., Perla, I., Shivanand, A. V.: Customer Segmentation Using K Means Algorithm and RFM Model. In: International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pp. 1018-1023 (2023).

[7] Li, P., Wang, C., Wu, J., Madleňák, R.: An E-commerce Customer Segmentation Method based on RFM Weighted K-means. In: International Conference on Management Engineering, Software Engineering and Service Sciences (ICMSS), pp. 1-6 (2022).

[8] Azadnia, A. H., Saman, M. Z. M., Wong, K. Y., Hemdi, A. R.: Integration model of Fuzzy C means clustering algorithm and TOPSIS Method for Customer Lifetime Value Assessment. In: IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 606-610 (2011).

[9] Husnah, M., Novita, R.: Clustering of Customer Lifetime Value With Length Recency Frequency and Monetary Model Using Fuzzy C-Means Algorithm. In: International Conference on Informatics Electrical and Electronics (ICIEE), pp. 1-6 (2022).

[10] Parkhan, A., Purnomo, M. R. A.: Industrial Marketing Strategy Formulation Based on RFM-Data Analytics and Fuzzy Database. In: International Conference On Cyber Management And Engineering (CyMaEn), pp. 1-7 (2023).

[11] Pramono, P. P., Surjandari, I., Laoh, E.: Estimating Customer Segmentation based on Customer Lifetime Value Using Two-Stage Clustering Method. In: 16th International Conference on Service Systems and Service Management (ICSSSM), pp. 1-6 (2019).

[12] Pal, N. R., Pal, K., Keller, J. M., Bezdek, J. C.: A possibilistic fuzzy c-means clustering algorithm. IEEE Transactions on Fuzzy Systems, vol. 13, no. 4, pp. 517-530 (2005).

[13] Monalisa, S.: Analysis Outlier Data on RFM and LRFM Models to Determining Customer Loyalty with DBSCAN Algorithm. In: International Symposium on Advanced Intelligent Informatics (SAIN), pp. 133-138 (2018).

[14] Bezdek, J. C.: Cluster validity with fuzzy sets. Journal of Cybernetics, vol. 3, no. 3, pp. 58-73 (1974).

[15] Fukuyama, Y., Sugeno, M.: A new method of choosing the number of clusters for the fuzzy c-means method. In: Proceedings of 5th Fuzzy Systems Symposium, pp. 247-252 (1989).

[16] Xie, X. L., Beni, G.: A Validity Measure for Fuzzy Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 8, pp. 841-847 (1991).

[17] Guo, W., Lin, K., Ye, W.: Deep Embedded K-Means Clustering. arXiv preprint arXiv:2109.15149 (2021).

[18] Dunn, J. C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics, vol. 3, no. 3, pp. 32-57 (1973).

[19] Lloyd, S.: Least squares quantization in pcm. IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129-137 (1982).

[20] Bishop, C. M.: Pattern recognition and machine learning. Springer (2006).

[21] Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: Proceedings of the 33rd International Conference on Machine Learning (ICML), PMLR, vol. 48, pp. 478-487 (2016).

[22] Guo, X., Gao, L., Liu, X., Yin, J.: Improved deep embedded clustering with local structure preservation. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1753-1759 (2017).

[23] Guo, X., Liu, X., Zhu, E., Yin, J.: Deep clustering with convolutional autoencoders. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 30 (2017).

[24] Yang, B., Fu, X., Sidiropoulos, N. D., Hong, M.: Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In: Proceedings of the 34th International Conference on Machine Learning (ICML), PMLR, vol. 70, pp. 3861-3870 (2017).

[25] Fard, M. M., Thonet, T., Gaussier, E.: Deep k-means: Jointly clustering with k-means and learning representations. Pattern Recognition Letters, vol. 138, pp. 185-192 (2020).

[26] Ghasedi Dizaji, K., Herandi, A., Deng, C., Cai, W., Huang, H.: Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 5736-5745 (2017).

[27] Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5147-5156 (2016).

[28] Zhang, R., Tong, H., Xia, Y., Zhu, Y.: Robust embedded deep k-means clustering. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM), pp. 1181-1190 (2019).

[29] Kingma, D. P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2015). (arXiv:1412.6980)

[30] Krizhevsky, A., Sutskever, I., Hinton, G. E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 25 (2012).

[31] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS), JMLR Workshop and Conference Proceedings, vol. 9, pp. 249-256 (2010).

[32] Ding, C., He, X.: K-means clustering via principal component analysis. In: Proceedings of the 21st International Conference on Machine learning (ICML), p. 29 (2004).

[33] Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research (JMLR), vol. 9, no. Nov, pp. 2579-2605 (2008).

[34] Vinh, N. X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research (JMLR), vol. 11, no. Oct, pp. 2837-2854 (2010).