Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Efficient HMM-Based Interpretable Customer Representation for Customer Segmentation in Banking

KELY MAXIME MOTUE DJOKO

University of Yaoundé 1, Faculty of Sciences

SYLVAIN ILOGA

sylvain.iloga@gmail.com

University of Maroua, Higher Teachers' Training College

Research Article

Keywords:

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10621937/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Efficient HMM-Based Interpretable Customer Representation for Customer Segmentation in Banking

KELY MAXIME MOTUE DJOKO[1,2,3†] and SYLVAIN ILOGA[3,4,5*†]

[1]Department of Computer Science, University of Yaoundé 1, Faculty of Sciences, Yaoundé, Cameroon.
[2]Fondation pour la Recherche, l'Ingénierie et l'Innovation, FR2I, Yaoundé, 14306, Cameroon.
[3]IRD, UMMISCO, University of Sorbonne, Bondy, F-93143, France.
[4*]Department of Computer Science, University of Maroua, Higher Teachers' Training College, Maroua, Cameroon.
[5]ETIS UMR 8051, CY Cergy Paris University, ENSEA, CNRS, Cergy, F-95000, France.

*Corresponding author(s). E-mail(s): sylvain.iloga@gmail.com;
Contributing authors: motuekely@gmail.com;
†These authors contributed equally to this work.

Abstract

Customer segmentation is a fundamental component of banking decision-support systems, supporting applications such as targeted marketing, risk management, fraud detection, and strategic planning. Existing approaches generally combine feature extraction with clustering, relying on either static or dynamic *Recency–Frequency–Monetary* (RFM) representations or deep learning models. However, RFM-based methods often fail to capture the sequential dynamics of customer behavior, while deep learning approaches typically produce latent representations that are difficult to interpret and frequently require additional dimensionality reduction for visualization. To address these limitations, this paper introduces the *Sequential Amounts and Sequential Intensity* (SASI) framework, an interpretable customer representation method based on Hidden Markov Models (HMMs). Unlike conventional deterministic approaches, HMMs provide a probabilistic framework for modeling sequential variations in transaction amounts and transactional intensity. The meta-data extracted from the learned HMMs are transformed into a two-dimensional embedding, whose axes directly reflect these two behavioral dimensions. Experiments conducted on **1,503** and **2,039** customers from the synthetic *Nigerian Retail Banking Transactions (Nigerian)* and the real-world *PKDD'99 (Berka)* datasets demonstrate the effectiveness of SASI using the *Silhouette Coefficient* (SC), *Davies–Bouldin Index* (DBI), and *Calinski–Harabasz Index* (CHI). When SASI is compared with the competing methods, it achieves the second-best **SC** (**0.650635**), the best **DBI** (**0.418718**), and the best **CHI** (**6578.11**) on the *Nigerian* dataset. On the *Berka* dataset, it attains the best **SC** (**0.507139**), the best **DBI** (**0.500209**), and the second-best **CHI** (**3309.01**). These results establish SASI as an effective operational framework, combining competitive clustering performance with fully interpretable and directly visualizable customer representations.

Keywords: Customer segmentation, RFM representations, Hidden Markov Models, Deep learning models

1 Introduction

Customer segmentation is a fundamental component of modern banking decision-support systems because it supports personalized marketing, credit risk assessment, fraud detection, and strategic planning [1, 2]. The rapid expansion of digital financial services, particularly in emerging economies, has considerably increased the volume and diversity of banking transactions, making accurate customer representation a key challenge for financial institutions [3, 4]. Beyond improving business performance, customer segmentation contributes to transparent and data-driven decision making, which is increasingly required for regulatory compliance and trustworthy artificial intelligence in banking [5].

Most existing customer segmentation frameworks follow a two-stage pipeline composed of *feature extraction* and *clustering*. During the clustering step, various clustering algorithms that can handle static or time series data are used to organize customers into compact and distinct clusters, according to their features. These customer features are either manually engineered from domain knowledge or automatically learned from transactional data. Among automatic approaches, *Recency–Frequency–Monetary* (RFM) representations remain the most widely adopted because of their simplicity and interpretability [6–8]. More recently, dynamic RFM variants and deep learning models such as *Variational Autoencoders* (VAEs), *Long Short-Term Memory* (LSTM) networks and *Convolutional Neural Networks* (CNNs) have been proposed to capture temporal customer behavior [9–12]. Although these approaches often improve predictive performance, they generally produce high-dimensional latent representations that remain difficult to interpret and typically require additional dimensionality reduction techniques for visualization.

Despite their popularity, existing feature extraction techniques only partially characterize transactional behavior. In particular, RFM-based methods summarize customer activity through aggregated statistics while ignoring the sequential organization of transactions within an observation window. Consequently, customers exhibiting similar recency, frequency and monetary values may nevertheless follow markedly different behavioral dynamics.

Figure 1 illustrates this limitation using three customers sharing identical recency and transaction frequency. Although indistinguishable from the perspective of conventional RFM representations, their transaction sequences reveal three distinct temporal behaviors corresponding to early, late and uniformly distributed activity. Such sequential information is ignored by existing RFM descriptors, although it may considerably influence customer similarity.

A similar limitation arises for the monetary dimension. Table 1 shows three customers with substantially different transaction magnitudes. While traditional monetary indicators distinguish customers only through aggregated values such as total amount or net flow, the normalized transaction sequences reveal that customers U and W exhibit identical sequential monetary behaviors despite their different financial scales, whereas customer V follows an opposite pattern. Capturing these sequential dynamics is therefore essential for obtaining behaviorally meaningful customer representations.

To address these limitations, this paper proposes the *Sequential Amounts and Sequential Intensity* (SASI) framework, an interpretable customer representation based on *Hidden Markov Models* (HMMs). Instead of relying on handcrafted aggregate statistics, SASI models customer behavior directly from transaction histories by jointly learning the sequential evolution of transaction amounts and transaction intensity. Meta-information extracted from the learned HMMs is then transformed into two interpretable behavioral scores defining a low-dimensional customer embedding. Unlike deep latent representations, the proposed embedding can be directly visualized, interpreted by banking analysts and used with conventional clustering algorithms without requiring additional dimensionality reduction.

The main contributions of this paper are fourfold:

1. We propose an interpretable HMM-based customer representation that jointly captures sequential transaction amounts and transaction intensity using transaction histories only.
2. We introduce a two-dimensional behavioral embedding that preserves semantic interpretability while providing a direct visual representation of customer behavior.

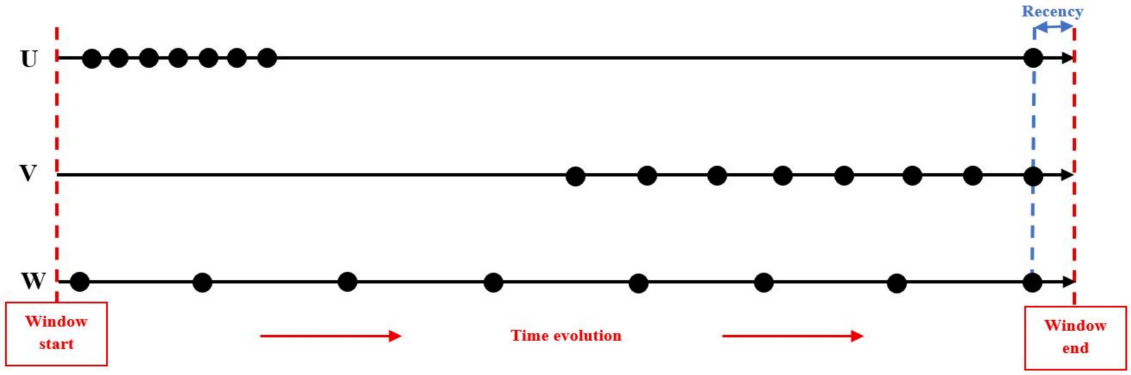


Figure 1: Three customers U, V and W, with identical frequency and recency, but different sequential transactional intensities within a window.

Table 1: Three customers U, V and W, with different monetary, but identical sequences of transactional normalized amounts within a window. m_i is the amount of the i^{th} transaction such that deposits are positive and withdrawals are negative, with $(1 \leq i \leq 8)$.

a. Original amounts

	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	**Total amount**	**Net flow**
U	−200	750	−800	1000	500	−125	100	50	3525	1275
V	250	500	−625	2500	5000	−4000	3750	−1000	17625	6375
W	−2000	7500	−8000	10000	5000	−1250	1000	500	35250	12750

b. Normalized amounts

	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	**Maximum amount**
U	−20%	75%	−80%	100%	50%	−12.5%	10%	5%	1000
V	5%	10%	−12.5%	50%	100%	−80%	75%	−20%	5000
W	−20%	75%	−80%	100%	50%	−12.5%	10%	5%	10000

3. The proposed representation remains fully compatible with conventional clustering algorithms, enabling its seamless integration into existing customer segmentation pipelines.
4. We demonstrate, through experiments conducted on two open datasets that SASI provides an effective and computationally efficient framework for customer segmentation in banking applications.

The remainder of this paper is organized as follows. Section 2 reviews related work on customer segmentation and customer representation. Section 3 briefly introduces the HMM background. Section 4 presents the proposed SASI framework, while Section 5 reports the experimental evaluation. Finally, the last section concludes the paper and discusses future research directions.

2 Background

Customer segmentation generally relies on two successive stages: *feature extraction* and *clustering*. The first stage transforms raw transactional records into feature vectors describing customer behavior, whereas the second stage groups customers with similar behavioral characteristics. Consequently, the quality of the final segmentation strongly depends on the descriptive power, interpretability and computational efficiency of the adopted feature extraction technique.

2.1 Feature extraction

Feature extraction can be performed either manually, through the design of handcrafted features, or automatically using deep learning models or

RFM-based representations.. Their respective objectives are to summarize customer behavior while preserving information relevant for downstream clustering and decision-support tasks.

2.1.1 Static RFM

The *Recency–Frequency–Monetary* (RFM) representation remains the most widely adopted automatic customer representation because of its simplicity, interpretability and computational efficiency [6–8]. It summarizes each customer by three complementary indicators describing the time elapsed since the last transaction (Recency), the number of transactions (Frequency), and the aggregated monetary value of transactions (Monetary). Owing to its low dimensionality, static RFM has been successfully applied to customer segmentation, customer valuation and targeted marketing [13–17].

Despite these advantages, static RFM aggregates all transactions within a single observation period and therefore ignores temporal dynamics. Customers having identical RFM triplets may nevertheless exhibit markedly different behavioral evolutions, as illustrated in Figure 1 and Table 1. Consequently, static RFM cannot distinguish customers whose transaction intensity or spending patterns evolve differently over time.

2.1.2 Dynamic RFM

Dynamic RFM representations extend the classical model by computing RFM descriptors over successive sliding windows, thereby generating multivariate time series that describe the temporal evolution of customer activity [18–21]. This strategy improves the detection of behavioral trends, seasonal effects and abrupt changes that remain invisible to static representations.

Several studies have further combined dynamic RFM with sequential learning techniques such as LSTMs or CNNs to improve customer profiling and segmentation [11, 12]. Nevertheless, dynamic RFM still relies on aggregated statistics (e.g., minimum, maximum, mean, variance, slope, etc.) computed within each window. Equation 1 shows how to compute some aggregated features for a time series g.

$$
\begin{aligned}
&\textit{min}: && \min_g = \min\{g(t) \mid 1{\leq}t{\leq}n\} \\
&\textit{max}: && \max_g = \max\{g(t) \mid 1{\leq}t{\leq}n\} \\
&\textit{mean}: && \mu_g = \frac{1}{n}\sum_{t=1}^{n} g(t) \\
&\textit{variance}: && \sigma_g^2 = \frac{1}{n}\sum_{t=1}^{n} (g(t)-\mu_g)^2 \\
&\textit{slope}: && \begin{cases} \beta_g = \dfrac{\sum_{t=1}^{n}(t-\mu_t)(g(t)-\mu_g)}{\sum_{t=1}^{n}(t-\mu_t)^2} \text{ where} \\ \mu_t = \left(\dfrac{1}{n}\sum_{t=1}^{n} t\right) = \dfrac{n+1}{2} \end{cases}
\end{aligned}
\tag{1}
$$

As a result, it captures the evolution of RFM indicators but not the sequential organization of transaction amounts and transaction intensity inside individual windows.

2.1.3 RFM extensions

To enrich customer descriptions, several extensions of the classical RFM model have been proposed by incorporating additional temporal, categorical, demographic or application-specific attributes. Representative examples include LRFMP [22], RFMC [23], RFMT [24], RFMD [25, 26] and RFME [27]. These models improve descriptive power by integrating complementary customer information beyond the original (R, F, M) triplet.

Although extended RFM representations often enhance segmentation quality, they also increase feature dimensionality, require additional feature engineering and may reduce interpretability. More importantly, they still describe customer behavior through aggregated descriptors and therefore remain unable to characterize the sequential evolution of transaction intensity and transaction amounts within an observation window.

2.1.4 Deep learning representations

Recent advances in deep learning have introduced powerful representation learning techniques for customer analytics. Variational Autoencoders (VAEs) learn compact latent embeddings capable of preserving complex nonlinear relationships and have demonstrated promising results for banking customer segmentation [9, 10]. More recently,

Transformer-based models have been employed to model long-range dependencies in customer interaction sequences [28], whereas Graph Neural Networks (GNNs) have exploited relational information for interaction prediction and churn analysis [29, 30].

Compared with RFM-based approaches, deep learning models generally provide richer representations and greater predictive capability. However, they often require substantial computational resources, extensive hyperparameter tuning and large training datasets. In addition, the learned latent representations usually remain difficult to interpret from a business perspective, thereby limiting their adoption in transparent banking decision-support systems. These limitations motivate the development of customer representations that simultaneously preserve behavioral expressiveness, computational efficiency and interpretability.

2.2 Clustering algorithms

Once customer representations have been obtained, clustering algorithms are employed to identify groups of customers exhibiting similar behavioral characteristics. Existing methods can be broadly categorized into five families according to their underlying grouping principles: *partition-based*, *density-based*, *hierarchical*, *graph-based*, and *time-series similarity-based* clustering [8].

Partition-based methods, including *K–means*, *K–medoids*, and *Gaussian Mixture Models* (GMMs), remain the most widely adopted approaches because of their simplicity and effectiveness for low-dimensional customer representations [6, 10, 14]. Density-based methods such as *DBSCAN* and *HDBSCAN* automatically identify arbitrarily shaped clusters, while explicitly handling noise [13, 31]. Hierarchical clustering constructs multi-level customer taxonomies without requiring a predefined number of clusters, whereas graph-based and spectral clustering exploit similarity graphs to capture complex nonlinear relationships [17]. Finally, algorithms such as *K–shape* are specifically designed for clustering behavioral time series according to shape similarity [20].

The choice of a clustering algorithm depends on the dimensionality of the representation, the structure of the data and the desired trade-off between computational efficiency, robustness and interpretability. Since the objective of the present work is to propose an interpretable customer representation rather than a new clustering technique, the proposed SASI framework is designed to remain independent of the clustering algorithm and can therefore be combined with conventional unsupervised methods. In the experimental evaluation, K–means is adopted because of its computational efficiency, widespread use in customer segmentation, and straightforward interpretation of the resulting customer groups.

2.3 Related studies

Table 2 synthesizes the main information on the studies related to customer segmentation reviewed in Section 2. The performances of these studies are generally evaluated using widespread metrics such as the *Silhouette coefficient* (SC), the *Davies-Bouldin index* (DBI) and the *Calinski-Harabasz index* (CHI). A technical presentation of these metrics is available in Section 5.3. Complementary related studies are analyzed in the surveys proposed in [6–8] and Figure 2 depicts the global taxonomy of customer segmentation used in the related papers reviewed in the current work, including the feature extraction and the clustering steps.

2.4 Problem statement

The previous review highlights two complementary trends in customer representation learning. RFM-based methods remain attractive because they are simple, interpretable and computationally efficient, but they rely on aggregated descriptors that ignore the sequential organization of transactions within an observation window. Conversely, deep learning approaches capture complex behavioral patterns but usually generate latent representations that are difficult to interpret and computationally demanding.

Consequently, an important gap remains between these two families of approaches. Existing methods do not simultaneously provide:

1. an interpretable customer representation derived directly from transaction histories;
2. a low-dimensional embedding suitable for direct visualization;

Table 2: Performance of relevant studies related to customer segmentation reviewed in Section 2. The performances are in column 6 where K denotes the final number of clusters.

Features	Authors[Ref]	Year	Data source	Clustering algorithms	Perf.	Specific Drawbacks
Static RFM-based	Brahmana et al. [13]	2020	Sales transaction data of PT. Cimory. From Jan. 2013 to Dec. 2018. 1661 customers.	K–means, K-medoids, DBSCAN	(K=2), SC: 0.9126, DBI: 0.33	Ignores temporal dynamics
	Rajesh et al. [14]	2024	E-commerce customer sales dataset from Kaggle. From 09/08/2020 to 26/10/2020.	K–means, Hierarchical clustering	(K=3), SC: 0.4389	
	Rusly et al. [15]	2025	PT Karya Logistik. dataset from Kaggle. From Jan. 2022 to Dec. 2024. 400 customers.	K–means	(K=5), SC: 0.4103	
	Grigorova et al. [16]	2025	Private financial dataset. From 01/07/2022 to 01/07/2024	K–means, Hierarchical clustering	(K=7), SC: 0.452, DBI: 0.810	
	Vianna et al. [17]	2025	Retail II dataset from UCI. From 01/12/2009 to 09/11/2011. 5878 customers	Graph-based clustering	(K=4), SC: 0.4289	
Dynamic RFM-based	Abbasimehr et al. [18]	2019	Private bank dataset. 7 months. 2531 customers.	Hierarchical clustering	(K=4), SC: 0.35	- Visualization issues - Increased complexity - Reduced interpretability - Sensitivity to l and κ
	Abbasimehr et al. [19]	2020	Private bank dataset. 11 months.	Hierarchical clustering	(K=4), SC: 0.34	
	Barkhordar et al. [11]	2021	Beka financial dataset. 5000 customers.	K–means	(K=3), SC: 0.862, DBI: 0.381	
	Abbasimehr et al. [20]	2022	Private bank dataset.	Hierarchical clustering, Spectral clustering, K-shape	(K=4), SC: 0.46, CHI: 5696.5	
	Wang et al. [21]	2024	Three datasets from ASP/SaaS-based manufacturing industry. From 2019 to 2021.	Spectral clustering, K-medoids, Hierarchical clustering	(K=4), SC: 0.9313, DBI: 0.0346	
VAE-based	Mancisidor et al. [9]	2018	Norwegian car loan real-word dataset from the Santander Consumer Bank Nordics. 12 months	K–means, Hierarchical clustering, HDBSCAN	(K=5), SC≈1.3, DBI≈1.8, CHI≈1.2	- Difficult to interpret - Depend on hyperparameter tuning and space regularization - Less explainable
	Mancisidor et al. [10]	2021	Norwegian car loan real-word dataset from the Santander Consumer Bank Nordics. 12 months	K–means, Hierarchical clustering, GMMs	(K=5), SC: –	

3. compatibility with conventional clustering algorithms;
4. computational efficiency compatible with large-scale banking applications.

To address these limitations, this work adopts HMMs as the underlying sequential modeling framework. HMMs constitute a well-established probabilistic model for sequential data and benefit from mature learning and inference algorithms whose robustness and efficiency have been extensively demonstrated. They have successfully been applied to customer relationship analysis [32, 33] and customer behavior prediction in e-commerce [34]. However, to the best of our knowledge, their potential for learning an interpretable dual representation of banking customers based jointly on transaction amount dynamics and transaction intensity dynamics has not yet been investigated.

Compared with alternative sequential models such as LSTMs, Transformers and VAEs, HMMs offer several advantages that are particularly relevant for banking applications. Their probabilistic formulation provides explicit latent behavioral

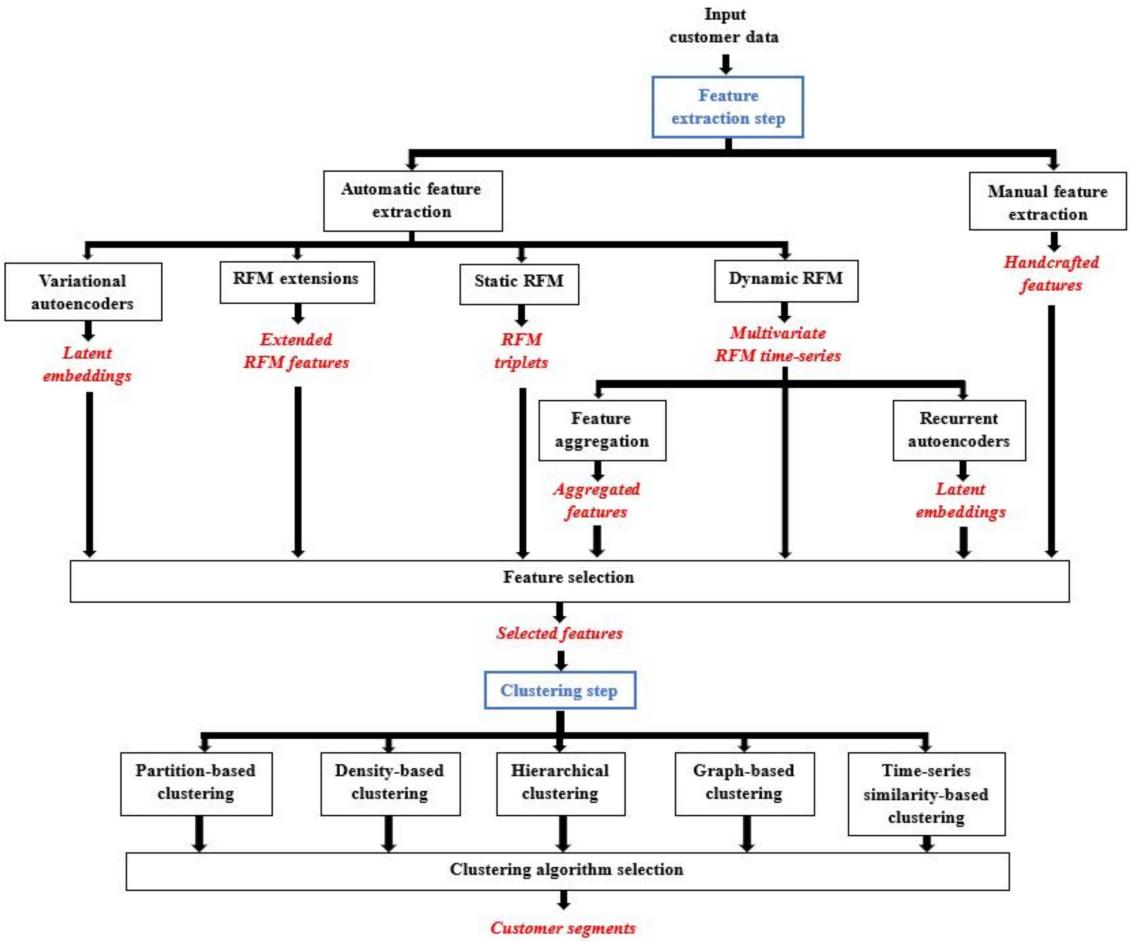


Figure 2: Global taxonomy of customer segmentation including the feature extraction and the clustering steps, respectively reviewed in Sections 2.1 and 2.2.

states, transition probabilities and emission distributions that can be directly interpreted by domain experts, thereby enhancing transparency and explainability. In addition, HMMs are generally more sample-efficient and computationally lighter than deep neural architectures, making them well suited to moderate-sized datasets and variable-length transaction sequences. These characteristics are consistent with the objectives of the proposed SASI framework, which seeks to capture sequential customer behavior while preserving interpretability, computational efficiency and compatibility with conventional customer segmentation pipelines.

The proposed SASI framework builds upon these properties by learning customer behavioral dynamics through HMMs and transforming the resulting model parameters into a compact two-dimensional representation that remains directly interpretable by banking analysts while being readily exploitable by standard clustering algorithms.

3 Hidden Markov models

This section briefly recalls the HMM concepts required by the proposed SASI framework. More comprehensive presentations can be found in [35].

3.1 Definition of a HMM

A HMM $\lambda = (A, B, \pi)$ is defined by:

1. a finite set of hidden states $S = \{s_1, \ldots, s_N\}$;
2. a finite observation alphabet $\vartheta = \{z_1, \ldots, z_M\}$;

3. a state transition matrix A, where $A[s_x, s_y] = Prob(q_{t+1} = s_y | q_t = s_x)$;
4. an observation probability matrix B, where $B[s_x, z_k] = Prob(z_k \text{ at time } t | q_t = s_x)$;
5. an initial state distribution π, where $\pi[s_x] = Prob(q_1 = s_x)$.

Each row of A, B and π is a probability distribution.

3.2 Notion of Markov chain

Given λ, an observation sequence $O = o_1 \ldots o_T$ is generated by iteratively selecting an initial state according to π, emitting one observation according to B, and moving to a new state according to A. The corresponding *Markov chain* is illustrated in Figure 3.

$$\begin{array}{rccccc} \textbf{\textit{symbols}}: & o_1 & & o_2 & \ldots & o_T \\ & \uparrow & & \uparrow & \vdots & \uparrow \\ \textbf{\textit{states}}: & q_1 & \rightarrow & q_2 & \rightarrow \ldots \rightarrow & q_T \end{array}$$

Figure 3: A Markov chain

3.3 Learning and inference

Given an observation sequence O, the likelihood $Prob(O|\lambda)$ is efficiently computed by the *Forward–Backward* algorithm [35] [1], whose complexity is $\mathcal{O}(T.N^2)$. Model parameters are estimated using the *Baum–Welch* algorithm, an *Expectation–Maximization* (EM) procedure [36] that iteratively maximizes the observation likelihood. Its computational complexity is $\mathcal{O}(\eta.T.N^2)$, where η denotes the maximum number of iterations.

3.4 Stationary state distribution

The vector $\varphi = (\varphi[s_1], \ldots, \varphi[s_N])$ is the stationary state distribution of a HMM $\lambda = (A, B, \pi)$ if [37] [2]:

1. φ is a probability distribution.
2. $(\varphi = \varphi.A)$.

$\varphi[s_x]$ estimates the overall proportion of time spent by λ in state s_x after a sufficiently long time. φ can be extracted from any row of the matrix $A^r = A \times A \times \ldots \times A$ (r times) when $r \rightarrow +\infty$. Its calculation requires $\mathcal{O}(r.N^3)$ operations.

3.5 Stationary symbol distribution

The stationary symbol distribution $\overline{\varphi} = (\overline{\varphi}[z_1], \ldots, \overline{\varphi}[z_M])$ of a HMM $\lambda = (A, B, \pi)$ is derived from its stationary state distribution through the formula $(\overline{\varphi} = \varphi.B)$ [38, 39] [3]. For every symbol $z_k \in \vartheta$, the value of $\overline{\varphi}[z_k]$ is calculated by Equation 2.

$$\overline{\varphi}[z_k] = \left(\sum_{s_x \in S} \varphi[s_x] \times B[s_x, z_k] \right) \qquad (2)$$

$\overline{\varphi}[z_k]$ estimates the overall proportion of time spent by λ observing the symbol z_k, regardless of the state from which this observation took place, after a sufficiently long time. When φ has already been calculated, the calculation of $\overline{\varphi}$ requires $\mathcal{O}(N.M)$ operations.

3.6 Quality of HMM learning

The quality of parameter estimation in a HMM trained with the *Baum–Welch* algorithm depends on the ratio $\left(\frac{T}{N^2}\right)$, where T denotes the observation sequence length and N the number of hidden states. Since the transition matrix contains N^2 parameters, reliable estimation generally requires that the available observations largely exceed the number of transition parameters to estimate. Consequently, small values of $\left(\frac{T}{N^2}\right)$ often lead to sparse transition matrices, poorly estimated probabilities and unstable models [40, 41].

Although informative initialization strategies can substantially improve learning quality, the ratio $\left(\frac{T}{N^2}\right)$ remains a useful practical indicator for selecting an appropriate model complexity. In the context of statistically informative initialization, the following empirical guideline is adopted throughout this work:

1. $\left(\frac{T}{N^2} < 1\right)$: unreliable learning;
2. $\left(1 \leq \frac{T}{N^2} < 3\right)$: limited reliability;
3. $\left(3 \leq \frac{T}{N^2} < 7\right)$: acceptable learning;
4. $\left(\frac{T}{N^2} \geq 7\right)$: robust learning.

These thresholds provide practical guidance for balancing model complexity and the amount of

[1] See Section III-A of [35]
[2] See Definition 9.1 of [37]
[3] See Equation 9 of [38] and Equation 5 of [39]

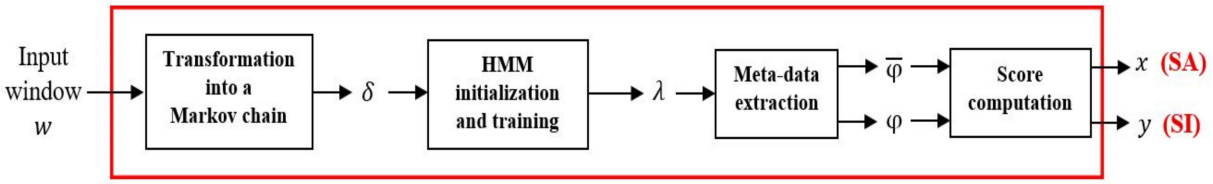


Figure 4: Methodology used in the **SASI** framework to associate a two-dimensional point (x, y) with each customer from the window w containing his chronological list of transactions.

$$\begin{array}{rl} \textbf{\textit{amounts}}: & a_1 \quad a_2 \quad \dots \quad a_T \\ & \uparrow \quad \uparrow \quad \vdots \quad \uparrow \\ \textbf{\textit{dates}}: & d_1 \rightarrow d_2 \rightarrow \dots \rightarrow d_T \end{array}$$

Figure 5: Pseudo MC $\hat{\delta}$ associated with customer U after the traversal of the window w containing his chronological list of transactions.

available data. For example, when $N = 7$, achieving at least an acceptable learning quality requires approximately ($T \geq 3 \times 7^2 = 147$) observations per customer.

4 The proposed approach

4.1 Main idea

Consider a bank customer U who has chronologically made T transactions during a fixed observation period. Assume also that each transaction is represented by the pair '*(date, amount)*', where the amount is positive for deposits and negative for withdrawals. More formally, customer U can be represented by the window containing his chronological list of transactions, that is $w=(d_1, a_1)(d_2, a_2)\dots(d_T, a_T)$, where d_i and a_i respectively denote the date and the amount of the i^{th} transaction. Our intuition is that U can be fully characterized by capturing the sequential variations of dates and amounts occurring in w_U. To achieve this goal, we consider that *customer U systematically traverses w from the first date d_1 to the last date d_T, observing each time the amount of the transaction realized at the current date d_i, before moving on to the next date d_{i+1}, until the observation of the amount associated with the last transaction*. This analysis enables to construct the pseudo MC $\hat{\delta}$ depicted in Figure 5, where states and symbols are respectively materialized by the dates and amounts.

$\hat{\delta}$ is not an effective MC because in a typical MC, states and symbols always belong to finite set. But this is not the case for the dates and the amounts in $\hat{\delta}$. Therefore, dedicated transformations will be applied to $\hat{\delta}$ to derive the effective MC δ associated with U, which will then be used to initialize and train a HMM λ that will capture the sequential amounts and the sequential intensity (date variation) of U. Meta-data extracted from λ will finally enable to derive reliable features to characterize U.

4.2 Methodology

4.2.1 SASI methodology

Figure 4 illustrates the methodology used in the proposed SASI framework to associate a two-dimensional point (x, y) with each customer, based on the window w containing his chronological list of transactions. This methodology is composed of the four following main steps:

1. **Transformation into a MC**: The content of the input window w is transformed into the MC δ, in order to capture the sequential variations of dates and amounts occurring in w.
2. **HMM initialization and training**: The statistical content of δ is used to initialize and train the HMM λ that will probabilistically learn the sequential variations of dates and amounts occurring in w.
3. **Meta-data extraction**: Meta-data characterizing the long-term behavior of the HMM λ are captured in the vectors φ and $\overline{\varphi}$.
4. **Score computation**: These meta-data are finally used to compute two scores x and y, respectively reflecting the sequential amounts (SA) and sequential intensity (SI) of the customer.

$$\begin{array}{ccccccc} 0 & & a_1 & \dots & a_T & & 0 \\ \uparrow & & \uparrow & \vdots & \uparrow & & \uparrow \\ d_{start} & \rightarrow & d_1 & \rightarrow \dots \rightarrow & d_T & \rightarrow & d_{end} \end{array}$$

(a) Initial pseudo MC $\hat{\delta}$

$$\begin{array}{ccccccc} \hat{a}_1 & & \hat{a}_2 & \dots & \hat{a}_T & & 0 \\ \uparrow & & \uparrow & \vdots & \uparrow & & \uparrow \\ \mathcal{D}(d_{start}, d_1) & \rightarrow & \mathcal{D}(d_1, d_2) & \rightarrow \dots \rightarrow & \mathcal{D}(d_{T-1}, d_T) & \rightarrow & \mathcal{D}(d_T, d_{end}) \end{array}$$

(b) Modified pseudo MC $\overline{\delta}$

$$\begin{array}{ccccccc} \overline{a}_1 & & \overline{a}_2 & \dots & \overline{a}_T & & z_0 \\ \uparrow & & \uparrow & \vdots & \uparrow & & \uparrow \\ \overline{\mathcal{D}}(d_{start}, d_1) & \rightarrow & \overline{\mathcal{D}}(d_1, d_2) & \rightarrow \dots \rightarrow & \overline{\mathcal{D}}(d_{T-1}, d_T) & \rightarrow & \overline{\mathcal{D}}(d_T, d_{end}) \end{array}$$

(c) Final MC δ

Figure 6: Step-by-step transformation of the window $\tilde{w}$ into the final MC δ.

The resulting embedding space of two-dimensional points associated with all the customers will finally be organized into compact and distinct segments during the clustering step.

4.3 Proposed SASI framework

Consider a customer represented by the chronological transaction window $w{=}(d_1, a_1)(d_2, a_2)\dots(d_T, a_T)$ where d_{start} and d_{end} denote the observation period boundaries. To account for the inactivity preceding the first transaction and following the last one, two dummy zero-amount transactions are inserted at the beginning and the end of the sequence. This produces the extended window $\tilde{w}$ presented in Equation 3 which preserves the customer's behavior, while ensuring that the complete observation period is represented.

$$\tilde{w} = (d_{start}, 0)(d_1, a_1)\dots(d_T, a_T)(d_{end}, 0) \quad (3)$$

4.3.1 Transformation into a MC

The extended window $\tilde{w}$ is converted into the pseudo MC $\hat{\delta}$ shown in Figure 6a, where dates define states and transaction amounts define observations. To model transactional intensity, each date (except d_{start}) is replaced by the corresponding inter-transaction delay, while transaction amounts are normalized by the maximum absolute amount as described in Equation 4.

$$\begin{cases} \hat{a_i} = 100{\times}\left(\dfrac{a_i}{a_{max}}\right) \text{ where} \\ \\ a_{max} = \max\{|a_i| \mid 1{\leq}i{\leq}T\} \end{cases} \quad (4)$$

This produces the modified MC $\overline{\delta}$ illustrated in Figure 6b. Normalized amounts are then discretized into $(2\mathcal{M}+1)$ intervals using Equation 5.

$$\begin{cases} z_0 = \{0\} \\ \\ z_k = \left]100{\times}\dfrac{(k-1)}{\mathcal{M}}, 100{\times}\dfrac{k}{\mathcal{M}}\right] \text{ with } (1{\leq}k{\leq}\mathcal{M}) \\ \\ z_{\mathcal{M}+k} = \left[-100{\times}\dfrac{k}{\mathcal{M}}, -100{\times}\dfrac{(k-1)}{\mathcal{M}}\right[\text{ with } (1{\leq}k{\leq}\mathcal{M}) \end{cases} \quad (5)$$

Each normalized amount $\hat{a}_i$ appearing in $\overline{\delta}$ is then replaced by the identifier $\overline{a}_i$ of its interval using Equation 6, this identifier is thus considered as a symbol of the HMM.

$$(\overline{a}_i = z_k) \Leftrightarrow (\hat{a}_i \in z_k), \text{ with } (0{\leq}k{\leq}2\mathcal{M}) \quad (6)$$

Inter-transaction delays are finally discretized into seven activity levels according to the clustering contained in Table 3, yielding the final MC δ illustrated in Figure 6c.

Table 3: Proposed clustering for the number of days separating two consecutive transactions in the **SASI** framework.

$\mathcal{D}(d_i, d_j)$ **in**	**Interpretation**	**Cluster** $\overline{\mathcal{D}}(d_i, d_j)$
$\{0\}$	The same day	s_0
$[1, 3]$	1 day to 3 days	s_1
$[4, 7]$	4 days to 1 week	s_2
$[8, 30]$	1 week to 1 month	s_3
$[31, 60]$	1 month to 2 months	s_4
$[61, 180]$	2 months to 6 months	s_5
$[181, +\infty]$	Beyond 6 months	s_6

Following the guideline introduced in Section 3.6, we recommend using at least T_{min}=150 transactions per customer, corresponding to the ratio ($\frac{T_{min}}{N^2} = \frac{150}{7^2} \approx 3.061$), which provides acceptable HMM learning reliability.

4.3.2 HMM initialization and training

Given a user-defined positive constant ε, the initial HMM $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$ associated with a customer is initialized from his final MC δ in order to preserve the empirical sequential dependencies.

1. The state set is $S = \{s_0, s_1, \ldots, s_{N-1}\}$, where N corresponds to the number of clusters used for transaction-delay clustering. In SASI, ($N = 7$) as defined in Table 3.
2. The symbol set is $\vartheta = \{z_0, z_1, \ldots, z_{2\mathcal{M}}\}$, where $\mathcal{M}$ is the parameter used in Equation 5. Consequently, the number of symbols is ($M = 2\mathcal{M} + 1$).
3. The transition probabilities are computed as described in Equation 7, where $transit(s_x, s_y, \delta)$ denotes the number of transitions from s_x to s_y in δ.

$$\overline{A}[s_x, s_y] = \frac{transit(s_x, s_y, \delta)}{transit(s_x, -, \delta) + \varepsilon} \tag{7}$$

4. The emission probabilities are computed as described in Equation 8, where $observe(z_k, s_x, \delta)$ represents the number of observations of symbol z_k from state s_x.

$$\overline{B}[s_x, z_k] = \frac{observe(z_k, s_x, \delta)}{observe(-, s_x, \delta) + \varepsilon} \tag{8}$$

5. The initial state probabilities are obtained from in Equation 9, where $start(s_x, \delta) = 1$ if δ starts with s_x, and 0 otherwise.

$$\overline{\pi}[s_x] = \frac{start(s_x, \delta)}{1 + \varepsilon} \tag{9}$$

The parameters of $\overline{\lambda}$ do not strictly form probability distributions because ε avoids zero denominators and null probabilities. Therefore, the missing probability mass is uniformly redistributed over each row as described in Equation 10, producing the adjusted initial HMM $\hat{\lambda} = (\hat{A}, \hat{B}, \hat{\pi})$, following the strategy adopted in [38, 39] [4].

$$\begin{cases} \hat{A}[s_x, s_y] = \overline{A}[s_x, s_y] + \dfrac{1}{N}\left(1 - \sum\limits_{h=1}^{N} \overline{A}[s_x, s_h]\right) \\ \hat{B}[s_x, z_k] = \overline{B}[s_x, z_k] + \dfrac{1}{M}\left(1 - \sum\limits_{h=1}^{M} \overline{B}[s_x, z_h]\right) \\ \hat{\pi}[s_x] = \overline{\pi}[s_x] + \dfrac{1}{N}\left(1 - \sum\limits_{h=1}^{N} \overline{\pi}[s_h]\right) \end{cases} \tag{10}$$

Finally, $\hat{\lambda}$ is trained using the *Baum–Welch* algorithm to learn the sequential patterns contained in δ and obtain the final customer-specific HMM $\lambda = (A, B, \pi)$.

Unlike generic initialization approaches (e.g., random, uniform, or K–means-based), this strategy directly exploits the sequential structure encoded in δ. It therefore provides a data-driven prior on state transitions and occupancies, reducing initialization ambiguity and improving the convergence behavior of the EM optimization process.

[4] See Equation 8 of [38] and Equation 4 of [39]

4.3.3 Meta-data extraction

The long-term behavioral characteristics of the trained HMM λ are summarized through two stationary distributions: the stationary symbol distribution $\overline{\varphi}$ presented in Section 3.5 and the stationary state distribution φ presented in Section 3.4. Choosing these two vectors to characterize each customer is very relevant in the banking context because they provide compact probabilistic descriptions of the customer's long-term transactional behavior. Indeed:

1. $\overline{\varphi}[z_k]$ is the overall proportion of time spent by the customer, making transactions whose normalized amounts belong to the sub-interval z_k as defined in Equation 5, regardless of the intensity with which these transactions were made, after a sufficiently long time.
2. $\varphi[s_x]$ is the overall proportion of time spent by the customer, making transactions with an intensity belonging to the cluster s_x as defined in Table 3, after a sufficiently long time.

4.3.4 Score computation

Assume that each customer has realized T transactions such that $(T_{min} \leq T \leq T_{max})$, where T_{min} and T_{max} are respectively the minimum and the maximum number of transactions considered for each customer. Only the T_{max} most recent transactions of customers for who $(T > T_{max})$ are considered.

Under these conditions, Equation 11 specifies the computation scheme of the final score x, associated with the customer's sequential amounts. Specifically, x is computed in Equation 11 as a weighted sum of the components of the vector $\overline{\varphi}$, where the weighting scheme distinguishes between transaction types: weights are set to zero for null amounts, positive for deposits, and negative for withdrawals. Moreover, these weights introduce a quadratic separation between two consecutive deposits z_i and z_{i+1}, with $(1 \leq i \leq \mathcal{M} - 1)$, and between two consecutive withdrawals z_j and z_{j+1}, with $(\mathcal{M} + 1 \leq j \leq 2\mathcal{M} - 1)$. This construction ensures a structured differentiation of transaction magnitudes within each category. The resulting sum is then scaled by the maximum transaction amount a_{max}, previously used in Equation 4 for normalization purposes. Finally, x is mapped onto a logarithmic scale to ensure scale invariance across customers with heterogeneous transaction volumes. This transformation also reduces the dominance of highly frequent symbols, preventing them from masking less frequent but behaviorally meaningful patterns. In addition, it provides an information-theoretic interpretation of the score in terms of surprise, thereby improving robustness to sampling noise in the estimated stationary distribution.

$$\begin{cases} x = \text{sign}(\theta^x) \times \log\left(1 + |\theta^x|\right) & \text{where} \\ \theta^x = a_{max} \times (\theta^+ + \theta^-) & \text{and} \\ \theta^+ = \left(\sum_{i=0}^{\mathcal{M}} i^2 \times \overline{\varphi}[z_i] \right) & \text{and} \\ \theta^- = \left(\sum_{j=\mathcal{M}+1}^{2\mathcal{M}} -(j - \mathcal{M})^2 \times \overline{\varphi}[z_j] \right) \end{cases} \tag{11}$$

Similarly, Equation 12 shows how to compute the final score y, associated with the customer's sequential intensity. The value of y is calculated in that equation as a weighted sum of the components of the vector φ, where each component is assigned a positive weight designed to enforce a quadratic separation between two consecutive states s_i and s_{i+1}, with $(0 \leq i \leq N-1)$. The resulting aggregated value is subsequently scaled by the total number T of transactions performed by the customer before to be mapped onto the logarithmic scale as it was the case for x. In particular, the value of y is finally moved to the interval $[0, 100]$ via the normalization with respect to the maximum possible score y_{max} which is reached when $(T = T_{max})$ and $(\varphi[s_0] = 1)$, all other components being equal to 0 (e.g., $\varphi = [\mathbf{1}, 0, 0, \ldots, 0,]$).

$$\begin{cases} y = 100 \times \left(\dfrac{\log\left(1 + \theta^y\right)}{y_{max}} \right) & \text{where} \\ \theta^y = T \times \left(\sum_{i=0}^{N-1} (N - i)^2 \times \varphi[s_i] \right) & \text{and} \\ y_{max} = \log\left(1 + T_{max} \times N^2\right) \end{cases} \tag{12}$$

The quadratic weighting applied in Equations 11 and 12 increases the contribution of distant symbols and states, thereby improving discrimination between regular and extreme customer behaviors. This choice is more appropriate than linear weighting, which underestimates large

deviations, while logarithmic weighting would reduce behavioral separability. The resulting scores provide compact and interpretable summaries of the probabilistic patterns captured by $\overline{\varphi}$ and φ.

4.3.5 Clustering step

The embedding space of two-dimensional points associated with all the customers resulting from the application of the proposed SASI methodology depicted in Figure 4 is finally organized into compact and distinct clusters through clustering experiments. The K–means algorithm has been selected for this purpose in the SASI framework because of its popularity and the quality of the results it produces.

4.3.6 Pseudocode of SASI

Algorithm 1 outlines the execution of the proposed SASI framework in a step-by-step manner. First, the number of symbols is computed (line 1). The algorithm then sequentially processes each customer (line 2) to compute the corresponding scores. For each customer, the associated sequence $w = (d_1, a_1)(d_2, a_2)\dots(d_T, a_T)$ is extracted (line 3) and used to build the corresponding Markov chain (MC) (line 4). This MC is subsequently used to initialize (line 5) and train (line 6) the final Hidden Markov Model (HMM) λ. The trained HMM is then used to compute the stationary distributions $\overline{\varphi}$ and φ (line 7), from which the scores (x, y) are derived (lines 8 and 9).

Algorithm 1 SASI

Inputs: $\{\text{Customers}, N, \mathcal{M}, \varepsilon, \eta, r\}$
Outputs: Customers

1: $M \leftarrow 2 * \mathcal{M} + 1$
2: **for each** $(c \in \text{Customers})$ **do**
3: $\quad w \leftarrow \text{ExtractSequence}(c.\text{dates}, c.\text{amounts})$
4: $\quad \delta \leftarrow \text{MarkovChainOf}(w)$
5: $\quad \hat{\lambda} \leftarrow \text{InitialHMM}(\delta, N, M, \varepsilon)$
6: $\quad \lambda \leftarrow \text{TrainHMM}(\hat{\lambda}, \delta.\text{symbols}, N, M, \eta)$
7: $\quad (\overline{\varphi}, \varphi) \leftarrow \text{StationaryDist}(\lambda, r)$
8: $\quad (x, y) \leftarrow \text{Scores}(\overline{\varphi}, \varphi, N, \mathcal{M})$
9: $\quad c.\text{scores} \leftarrow (x, y)$
10: **end for**
11: **return** Customers

4.3.7 Theoretical time cost

Suppose that the experimental dataset contains a total of $|\mathcal{C}|$ costumers, where $\mathcal{C}$ is the set of customers. The goal of this section is to evaluate the theoretical time cost of the main contribution of the current paper which is, the association of a two-dimensional point (x, y) with each customer from the window $w = (d_1, a_1)(d_2, a_2)\dots(d_T, a_T)$ containing his chronological list of transactions, following the proposed SASI methodology presented in Figure 4.

In the context of the proposed SASI where each customer is associated with one HMM λ composed of $(N{=}7)$ states and $(M{=}2\mathcal{M}{+}1)$ symbols, the theoretical time cost of the underlying methodology is evaluated as follows:

1. **Transformation into a MC**: This step is mainly dominated by the time required to read the customer data, augmented by the time needed to compute the number of days $\mathcal{D}(d_i, d_j)$ between two dates. All the remaining operations (normalization and clustering) are less time consuming. This step consequently runs in $\mathcal{O}(T_{max}.(b + \overline{b}))$, where b is the time required to read one transaction and $\overline{b}$ is the computation time of $\mathcal{D}(d_i, d_j)$.
2. **HMM initialization and training**: The time required for initializing a HMM was experimentally very low compared to the HMM training time. This step is therefore dominated by the HMM training phase, which runs in $\mathcal{O}(\eta.T_{max}.N^2)$ according to Section 3.3, where η is the user-defined maximum number of iterations of the *Baum–Welch* algorithm.
3. **Meta-data extraction**: The most time consuming operation realized during this step is the computation of the stationary state distribution φ, which runs in around $\mathcal{O}(r.N^3)$ as stated in Section 3.4, where r is the user-defined number of matrix products. This time cost is augmented by the time required for computing the stationary symbol distribution $\overline{\varphi}$, which is $\mathcal{O}(N.M)$ according to Section 3.5. This step thus runs in $\mathcal{O}(r.N^3 + N.M)$.
4. **Score computation**: In this step, $3M$ arithmetic operations are required for computing x, while $3N$ arithmetic operations are required for computing y. Consequently, this runs in $\mathcal{O}(N + M)$ since the computation time of the

logarithm function is very low compare to the remaining operations.

Finally, the overall time ξ required for computing the pair (x, y) for all the customers in the dataset is calculated by Equation 13.

$$\begin{cases} \xi = \mathcal{O}\left(|\mathcal{C}|.\left(\sum_{i=1}^{4} \xi_i\right)\right) & \text{where} \\ \xi_1 = T_{max}.(b + \bar{b}) & \text{and} \\ \xi_2 = \eta.T_{max}.N^2 & \text{and} \\ \xi_3 = r.N^3 + N.M & \text{and} \\ \xi_4 = N + M \end{cases} \tag{13}$$

5 Experimental results

This section aims to evaluate the performances of the proposed SASI framework through experiments realized on two open datasets following the methodology described in Figure 4. All the Python source codes and the experimental data supporting the findings of this study are available online [5].

5.1 Experimental datasets

The proposed framework was evaluated on the *Nigerian Retail Banking Transactions* dataset [42], that will be called the *Nigerian* dataset in the rest of the current paper for convenience. This recently released open dataset (October 2025) consists of synthetic retail banking transactions derived from current and savings accounts, and includes fraud detection labels. It captures realistic characteristics of Nigerian banking activity, such as the predominance of mobile banking, point-of-sale usage, and regional transaction patterns across states. The dataset has been designed to support multiple applications, including fraud detection and prevention, anomaly detection, channel optimization, customer segmentation, and risk modeling. Overall, the dataset contains 5,000,000 transactions collected over the period from d_{start}= 01/01/2023 to d_{end}= 31/12/2024, associated with 100,000 customers distributed across 14 banks and interacting with 24 merchants in 37 Nigerian states. The schema comprises 16 attributes, including the customer identifier in the strict format *CUS-????????* (where each *?* is a digit), the transaction amount expressed in *Nigerian Naira (NGN)*, and the transaction timestamp (Africa/Lagos time zone), which are essential variables for the implementation of the SASI framework.

SASI was also evaluated on the *PKDD'99 (Berka)* dataset [43]. This publicly available benchmark dataset originates from a Czech bank and has become one of the most widely used datasets for research on financial data mining and customer behavior analysis. It contains real-world banking operations associated with current accounts and has supported numerous studies in customer segmentation, credit risk assessment, fraud detection, recommendation systems, and financial behavior modeling. Overall, the dataset covers an observation period of six years, from d_{start}= 01/01/1993 to d_{end}= 31/12/1998. It contains transactional records associated with approximately 4,500 customers, including 682 labeled customers linked to loan records with known repayment status. All monetary values are expressed in *Czech koruna (CZK)*, while each customer is identified by a unique positive integer. The dataset comprises several relational tables describing customers, accounts, transactions, loans, cards, and demographic information. For the implementation of the proposed framework, only the customer identifier, transaction amount, and transaction date were retained, as these constitute the essential variables required to model sequential customer transactional behavior.

A filtering step was initially applied in each experimental dataset to retain only customers with a sufficient and consistent transaction history. Specifically, customers were selected if they performed T successful transactions such that ($150 \leq T \leq 500$), and if all transaction amounts were greater or equal to 12,500 *NGN* or 186 *CZK* (approximately 5,000 *XFA*), in order to reduce noise and focus on meaningful behavioral patterns. This selection resulted in a subset $\mathcal{C}$ of customers composed of ($|\mathcal{C}|$=1,503) and ($|\mathcal{C}|$=2,039) customers respectively for the *Nigerian* and the *Berka* datasets. Table 4 summarizes the main information related to the two experimental datasets.

[5] https://github.com/HmmBank/HMM-Based-SASI-Frameworks/

Table 4: Main information related to the experimental datasets respectively extracted from the *Nigerian Retail Banking Transactions* and the *Berka* datasets.

Data source	Data type	Observation period	Total customers	Extracted customers	T_{min}	T_{max}	Minimum amount
Nigerian [42]	Synthetic	2 years	100,000	1,503	150	500	12,500 *NGN*
Berka [43]	Real-word	6 years	4,500	2,039	150	500	186 *CZK*

5.2 Experimental settings

All experiments were conducted using *Python 3.9.11* on a personal computer running *MS Windows 10 Professional*, equipped with 16 GB RAM and an *Intel(R) Core(TM) i7-8665U* processor clocked at 1.90 GHz. . The parameter ($\mathcal{M}$=20) was used in Equation 5 to split the interval $[-100, 100]$, resulting in (M=41) observation symbols for each HMM. The number of hidden states was set to (N=7) and the maximum number of iterations of the *Baum–Welch* algorithm was fixed to (η=500). The stationary state distribution of each HMM was approximated using at most (r=1000) successive matrix multiplications, with an early stopping criterion to ensure computational efficiency. To improve numerical stability, the constant (ε=1) was added to the denominators of the initial HMM parameters in order to prevent divisions by zero and the occurrence of null probabilities following [38, 39] [6].

In this work, K–means clustering experiments (except for the combination with DTW) were carried out in the software *WEKA 3.9.3* [44] using the *SimpleKmeans* classifier with following hyperparameters: maxiterations=500, distance='*Euclidean*', initialization='*Canopy*', canopyT1=−1.25, canopyT2=−1.0. Unreported experiments realized with the *K–means++* initialization method exhibited less good performances.

5.3 Selected metrics of evaluation

To assess the quality of the obtained customer clusters, three widely adopted internal validation metrics are considered: the *Silhouette coefficient* (SC), the *Davies-Bouldin index* (DBI), and the *Calinski-Harabasz index* (CHI). These metrics evaluate cluster compactness and separation without requiring external labels, providing complementary assessments of segmentation quality.

[6] See Section VI-B of [38] and Section V-B of [39]

5.3.1 Silhouette coefficient

The SC measures the quality of individual cluster assignments by comparing the cohesion of each point within its own cluster and its separation from neighboring clusters. It is computed using Equation 14, where $\chi(i)$ is the average intra-cluster dissimilarity of point x_i, and $\overline{\chi}(i)$ is the minimum average dissimilarity between x_i and another cluster. The SC ranges from $[-1, 1]$: values close to 1 indicate well-separated clusters, values near 0 indicate overlapping clusters, and negative values suggest possible misassignments. Higher values therefore correspond to better clustering.

$$\begin{cases} \text{SC} = \dfrac{1}{|\mathcal{C}|}\left(\sum\limits_{i=1}^{|\mathcal{C}|} s(i)\right) \text{ where} \\ \\ s(i) = \dfrac{\overline{\chi}(i) - \chi(i)}{\max(\overline{\chi}(i), \chi(i))} \end{cases} \tag{14}$$

5.3.2 Davies-Bouldin index

The DBI evaluates the similarity between each cluster and its most similar cluster by jointly considering intra-cluster dispersion and inter-cluster separation. It is computed using Equation 15, where K is the number of clusters, σ_k represents the average intra-cluster distance of cluster c_k, and $dist(\hat{c}_k, \hat{c}_j)$ is the distance between cluster centroids. Equation 15 returns a value which belongs to $[0, +\infty[$ such that lower DBI values indicate compact clusters with better separation.

$$\begin{cases} \text{DBI} = \dfrac{1}{K}\left(\sum\limits_{k=1}^{K} \max\{R_{k,j} \mid j \neq k\}\right) \text{ where} \\ \\ R_{k,j} = \dfrac{\sigma_k + \sigma_j}{dist(\hat{c}_k, \hat{c}_j)} \end{cases} \tag{15}$$

5.3.3 Calinski-Harabasz index

The CHI, also called the *variance ratio criterion*, evaluates the global separation of a clustering by

comparing between-cluster and within-cluster dispersion. It is computed using Equation 16, where ρ and $\overline{\rho}$ denote respectively the between-cluster and within-cluster scatter matrices. Higher SC values indicate well-separated clusters with low internal dispersion.

$$\begin{cases} \text{CHI} = \dfrac{\text{Trace}(\rho)}{\text{Trace}(\overline{\rho})} \text{ where} \\ \\ \text{Trace}(X) = \sum_{i=1}^{m} X[i,i] \end{cases} \tag{16}$$

5.4 Customer segmentation results

The application of the proposed SASI framework to the customers extracted from the two experimental datasets produced a two-dimensional embedding space, in which each point characterizes a customer according to the sequential behavior of transactional amounts and transaction intensity. The *K–means* algorithm was subsequently applied using the optimal number of clusters, namely (K=5) for the two experimental dataset. This value was determined using the *elbow method* based on the within-cluster *Sum of Squared Errors* (SSE) reported in Table 5. The resulting customer segmentations are illustrated in Figures 7a and 7b.

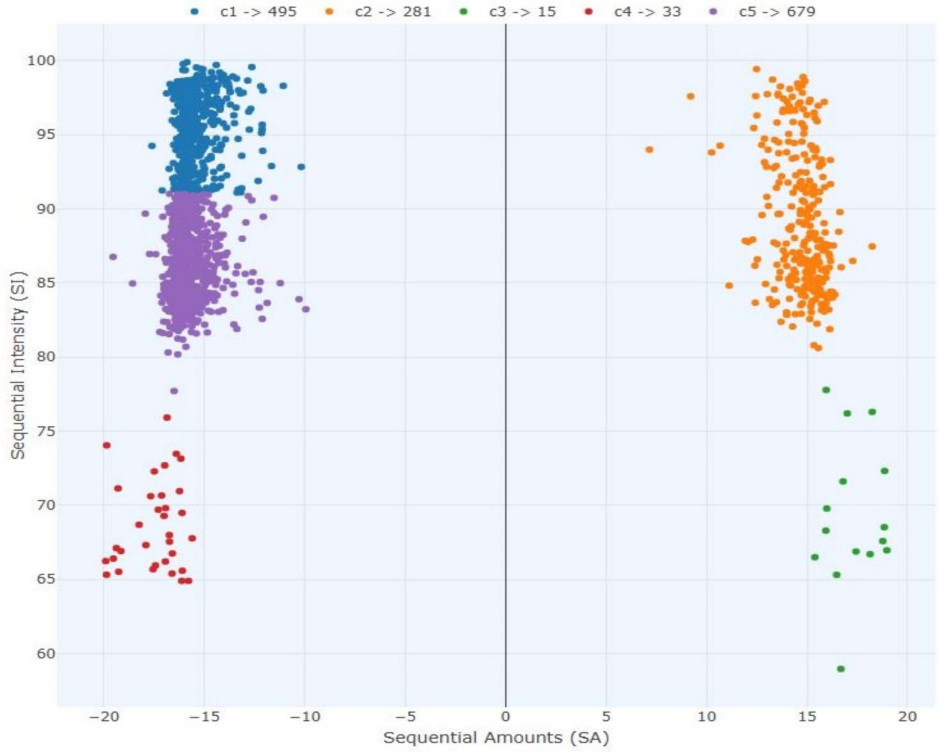
An important observation is that the segmentations shown in Figures 7a and 7b contain no duplicate representations, meaning that each customer is mapped to a unique pair of scores (x, y). This demonstrates the discriminative capability of the proposed SASI framework, which consistently assigns distinct representations to customers exhibiting different transactional sequences. Furthermore, the resulting embedding spaces remain fully interpretable from a banking perspective.

More specifically, both figures reveal a common hierarchical organization of customer behaviors. The first level of separation is governed by the score x, which reflects the sequential transaction amounts. This naturally partitions customers into two main groups located on opposite sides of the origin of the x-axis: *consumers*, characterized by negative values of x, and *contributors*, characterized by positive values. Although the value of x further differentiates customers within each main group, the second level of organization is primarily driven by the score y, which reflects the sequential transaction intensity. Consequently, each of the two main groups is further subdivided into two or three compact subgroups according to the intensity of customer activity. Depending on the range of y, these subgroups can be semantically interpreted as representing various categories of customers including *very intensive*, *intensive*, *regular*, *dormant*, etc. The final customer segments are therefore obtained by combining the behavioral interpretation associated with both axes, yielding intuitive and operationally meaningful profiles such as *very intensive consumer*, *regular contributor*, *dormant contributor*, etc.

Table 5: SSEs obtained by SASI in the two experimental datasets. The elbow is in bold.

K	**SSE in Nigerian**	**SSE in Berka**
2	36.45	17.79
3	27.35	15.81
4	23.94	14.68
5	**9.16**	**6.80**
6	6.26	4.97
7	5.83	4.23
8	5.63	3.52
9	3.35	3.01
10	2.64	2.70
11	2.24	2.44
12	2.06	2.27

The quality of the clusterings shown in Figures 7a and 7b was assessed using the three validation metrics introduced in Section 5.3. The proposed SASI framework achieved SC=0.650635, DBI=0.418718, and CHI=6578.11 on the *Nigerian* dataset, and SC=0.507139, DBI=0.500209, and CHI=3309.01 on the *Berka* dataset. These performances are summarized in Table 6. Overall, these results indicate that SASI produces high-quality customer segmentations on both datasets. The relatively high SC demonstrate strong intra-cluster cohesion together with good inter-cluster separation, suggesting that most customers are assigned to well-defined behavioral groups. This observation is reinforced by the low DBI, which indicate compact clusters that are clearly separated from one another. Furthermore, the high CHI reveal a favorable balance between between-cluster and



(a) *K–means* clustering of the *Nigerian* dataset with (K=5)



(b) *K–means* clustering of the *Berka* dataset with (K=5)

Figure 7: Customer segmentation of the two experimental datasets derived by the SASI framework.

within-cluster dispersion, confirming the overall consistency of the identified partitions.

Table 6: Performances of SASI on the two experimental datsets.

	Nigerian	**Berka**
SC	0.650635	0.507139
DBI	0.418718	0.500209
CHI	6578.11	3309.01

From a banking perspective, these results indicate that the derived customer segments are sufficiently distinct to support a wide range of decision-support applications, including targeted marketing, personalized product recommendation, customer relationship management, credit risk assessment, and behavioral analysis. The combination of high intra-cluster homogeneity, strong inter-cluster separation, and the intrinsic interpretability of the proposed SASI representation enhances the operational value of the resulting segmentation. To further facilitate the exploration of these results, an interactive HTML visualization of the clusterings presented in Figures 7a and 7b, generated by the proposed SASI framework, is publicly available online [5].

5.5 Experimental time cost

5.5.1 Sequential time cost

Table 7 reports the durations associated with the following main computational steps of the sequential implementation (executed on a single processor) of the proposed SASI framework across all experiments, in the two experimental datasets:

1. **Data access**: extraction of the transactions of all the customers from the dataset.
2. **HMM-based**: HMM-based computations.
3. **Scores**: computation of the pairs (x, y).
4. **Others**: remaining computations, including the edition of visualization files.

According to the content of Table 7, the proposed framework runs in reasonable time regarding the sizes of the experimental datasets. Indeed, around 1,075 seconds were required to process the 1,503 customers of the *Nigerian* dataset, while only 597 seconds were needed to process the 2,039 customers of the *Berka* dataset. In other words, an average of less than one second is needed in the proposed SASI framework for the processing of each customer. However, this time cost can rapidly increase as the dataset size grows. A step-by-step analysis of this time cost reveals that HMM-based computations are the most time consuming steps.

Table 7: Sequential time costs of SASI on the two experimental datsets. Durations are in seconds.

	Nigerian	**Berka**
Data access	19.71	7.90
HMM-based	1054.75	588.41
Scores	0.029	0.032
Others	0.011	0.002
Total	1074.5	596.34

Although HMM-based computations can become time-consuming for large-scale datasets, their computational complexity remains manageable through standard optimizations. It is in this perspective that parallel a version of the proposed framework has been implemented in order to reduce the total execution time, without compromising the quality of customer segmentation.

5.5.2 Parallel time cost

The parallel implementation was developed using the *Message Passing Interface* (MPI) [45, 46] for Python through the *mpi4py* library, and deployed on a star interconnect network composed of $\mathcal{P}$ computers (processors). The set of customers $\mathcal{C}$ was evenly partitioned across processors, such that each processor was assigned approximately $\frac{|\mathcal{C}|}{\mathcal{P}}$ customers, which it processed independently. Despite the fact that the number of transactions per customer is not unique, this data-parallel strategy attempts to ensure load balancing and to enable scalable execution of the proposed framework.

Only the HMM-based computations were parallelized. More precisely, a master-worker scheme was adopted in which the central processor (master) is responsible for reading the customer data and distributing disjoint subsets of customers to the worker processors. Each processor then performs the HMM-based computations on its local

Table 8: Experimental time costs of sequential and parallel implementations of the proposed SASI framework in the two experimental datasets. Durations are in seconds.

(a) *Nigerian* dataset

	$\mathcal{P}$	**Duration**	**Speed-up**
Sequential	1	1075	–
Parallel	4	512	2.099
	6	386	2.784
	8	346	3.106

(b) *Berka* dataset

	$\mathcal{P}$	**Duration**	**Speed-up**
Sequential	1	597	–
Parallel	4	260	2.296
	6	214	2.789
	8	191	3.125

subset. Upon completion, the worker processors return their partial results to the master, which subsequently aggregates them to compute the final scores and generate the visualization files in a centralized manner. Experiments were conducted for $\mathcal{P}\in\{4, 6, 8\}$. The fastest parallel implementations only required around 346 and 191 seconds respectively for the *Nigerian* and the *Berka* dataset, which respectively induced a best speed-up of ($\frac{1075}{346}\approx$3.106) and ($\frac{597}{191}\approx$3.125). Table 8 summarizes the time costs of sequential and parallel implementations of the proposed SASI framework.

Although the best empirical speed-up achieved by the SASI framework in each experimental dataset remains moderate, the reduction in computational time induced by the proposed parallel implementations is nevertheless significant. Moreover, this gain is expected to increase as the dataset size grows, owing to the data-parallel nature of the approach. Overall, these experimental findings indicate that the proposed SASI framework exhibit satisfactory scalability properties, making them suitable for large-scale customer data analysis in distributed computing environments.

5.6 Comparison to existing methods

5.6.1 Experimental comparison

The proposed SASI framework was compared with the three following representative customer representations on the two experimental datsets: static RFM, dynamic RFM (DRFM), and VAE-based representations. All methods were evaluated under identical clustering conditions. For the DRFM representation, a sliding window of length (l=180) days with an overlap of (κ=150) days was employed. This configuration generated 19 and 68 consecutive sub-windows for the *Nigerian* and *Berka* datasets, respectively. Consequently, each customer was represented by a multivariate time series of dimension 19×3=57 for the *Nigerian* dataset and 68×3=204 for the *Berka* dataset. The three following DRFM representations were investigated:

1. Raw multivariate time series clustered using K–*means-DTW*;
2. 15-dimensional aggregated features as defined in Equation 1 which computes five aggregated variables for each of the three time series;
3. Latent representations learned with an LSTM encoder using embedding dimensions of 2 and 15 (seed=42, epochs=100, batch size=32, optimizer=*Adam*).

Unlike the static customer vectors considered in [9, 10], SASI models variable-length transaction sequences. Consequently, their original MLP-based VAE architecture was replaced by a (Transformer/LSTM) VAE operating on sequences $\hat{w} = (0, a_1)(\mathcal{D}(d_1, d_2), a_2)\ldots(\mathcal{D}(d_{T-1}, d_T), a_T)$ using latent dimensions of 2 and 15 with seed=42, epochs=50, batch size=32, and the *Adam* optimizer.

For each representation, K–*means* clustering was performed after selecting the number of clusters with the elbow method. The corresponding SSE values are reported in Table 9, while the clustering performances are summarized in Table 10. PDF and interactive HTML visualizations of the clusterings produced by the static RFM, DRFM–LSTM$_2$, and VAE$_2$ approaches in each experimental dataset are publicly available in the online repository [5]. These visualizations have been omitted from the present paper owing to space constraints.

Table 9: SSEs obtained for the static RFM-based, the dynamic RFM-based and the VAE-based representations. The elbows are in bold.

(a) SSEs obtained in the *Nigerian* dataset

K	**Static RFM**	**Raw DRFM**	**DRFM & Aggreg.**	**DRFM & LSTM$_2$**	**DRFM & LSTM$_{15}$**	**VAE$_2$**	**VAE$_{15}$**
2	63.3	20.8	310.2	1.6794	439.3	10.36	93.85
3	34.6	19.9	177.7	**0.1104**	392.8	6.556	63.67
4	29.3	17.4	120.8	0.0822	338.5	3.368	54.46
5	20.9	17.0	116.2	0.07858	195.7	4.652	**48.45**
6	**17.9**	16.3	91.5	0.0764	**151.8**	**3.766**	45.23
7	16.3	16.0	**70.9**	0.0757	134.4	3.720	45.62
8	15.1	**15.0**	61.0	0.0755	123.0	2.978	42.03
9	14.2	14.9	55.9	0.0752	112.0	2.491	41.65
10	13.6	14.6	54.0	0.0751	105.0	2.333	41.64
11	13.2	14.2	51.5	0.0368	101.6	2.010	41.93
12	12.1	13.8	49.4	0.0316	94.56	2.089	39.71

(b) SSEs obtained in the *Berka* dataset

K	**Static RFM**	**Raw DRFM**	**DRFM & Aggreg.**	**DRFM & LSTM$_2$**	**DRFM & LSTM$_{15}$**	**VAE$_2$**	**VAE$_{15}$**
2	86.6	80.1	395.0	42.7	713.4	**0.760**	8.93
3	66.2	64.4	303.4	20.9	290.0	0.733	**3.04**
4	46.4	55.5	236.7	10.9	223.0	0.732	3.03
5	**44.9**	**51.3**	213.0	7.2	**203.3**	0.732	2.83
6	37.3	47.6	187.4	**4.4**	167.8	0.732	2.82
7	32.4	48.1	**166.2**	3.5	148.2	0.732	2.82
8	27.7	44.5	158.5	3.1	135.5	0.732	2.82
9	24.7	43.8	147.6	2.9	124.2	0.732	2.82
10	22.3	43.2	130.2	1.8	117.8	0.732	2.82
11	20.3	42.8	123.7	1.4	112.5	0.732	2.82
12	18.8	42.3	117.9	1.2	106.6	0.732	2.82

Table 10 indicates that the DRFM–LSTM$_2$ and VAE$_2$ representations achieve apparently excellent clustering metrics. However, these results are misleading because they are associated with a large number of duplicate embeddings, implying that many distinct customers are mapped onto identical latent representations. This degeneracy primarily results from excessive dimensionality reduction. The phenomenon is particularly pronounced for the VAE$_2$ model, where customer descriptions comprising up to (T_{max}=500) transaction pairs (i.e., 1000 input values) are compressed into a latent space of only 2 dimensions. Although VAE$_{15}$ does not generate duplicate embeddings in the *Berka* dataset, its clustering remains highly unbalanced, with one cluster containing 2036 of the 2039 customers, while the two remaining clusters contain only one and two customers, respectively. Consequently, the corresponding clustering performance cannot be considered meaningful. In contrast, DRFM–LSTM$_{15}$ generates only six duplicate embeddings in the *Nigerian* dataset and none in the *Berka* dataset, making its clustering results reliable.

Table 10 demonstrates that SASI provides good clustering performance, while preserving fully discriminative customer representations. In the *Nigerian* dataset, it achieves the second-best SC (0.650635), the best DBI (0.418718), and the best CHI (6578.11). Similarly, in the *Berka* dataset, it attains the best SC (0.507139), the best DBI (0.500209), and the second-best CHI

Table 10: Comparison between the performances of the SASI framework, the static RFM-based, the dynamic RFM-based and the VAE-based representations in the customer segmentation of the two experimental datasets. Invalid performances are colored in red. Feature extraction and clustering times are in seconds. Best valid performances are in bold.

(a) Comparisons in the *Nigerian* dataset

	Proposed SASI	**Static RFM**	**Raw DRFM**	**DRFM & Aggreg.**	**DRFM & LSTM_2**	**DRFM & LSTM_{15}**	**VAE_2**	**VAE_{15}**
Clusters	5	6	8	7	3	6	6	5
Maximum inputs	1000	1000	57	57	57	57	1000	1000
Latent dimension	2	3	–	15	2	15	2	15
Feature ext. time	1075	0.06	0.15	0.6	53	56	9598	8937
Clustering time	0.03	0.03	1145.5	0.06	0.01	0.02	0.01	0.04
Duplicates	0	0	0	0	274	6	1029	0
SC	0.650635	0.172121	0.0930921	0.130459	0.993042	0.346242	0.82212	**0.811669**
DBI	**0.418718**	1.85714	4.67872	9.01291	0.0682475	0.919282	0.982513	1.28074
CHI	**6578.11**	2842.91	137.567	841.604	2184250	2482.42	1454.78	4843.25

(b) Comparisons in the *Berka* dataset

	Proposed SASI	**Static RFM**	**Raw DRFM**	**DRFM & Aggreg.**	**DRFM & LSTM_2**	**DRFM & LSTM_{15}**	**VAE_2**	**VAE_{15}**
Clusters	5	5	5	7	6	5	2	3
Maximum inputs	1000	1000	204	204	204	204	1000	1000
Latent dimension	2	3	–	15	2	15	2	15
Feature ext. time	597	0.13	0.25	0.67	203.43	243.45	16770	6948
Clustering time	0.04	0.02	4013.17	0.03	0.03	0.02	0.04	0.03
Duplicates	0	0	0	0	45	0	1998	0
SC	**0.507139**	0.237677	0.154255	−0.0735237	0.802832	0.426085	0.997871	0.997986
DBI	**0.500209**	1.51992	2.14123	19.1128	0.446032	1.11583	0.980245	0.594307
CHI	3309.01	2259.62	334.452	539.334	67518	**5203.97**	1839.45	2787.24

(3309.01), confirming its robustness and effectiveness across both experimental datasets. Beyond these competitive clustering performances, SASI distinguishes itself by producing fully interpretable customer representations that can be directly visualized, thereby providing effective support for customer behavior analysis and banking decision-making.

Although SASI ranks second according to one evaluation metric on each experimental dataset, being outperformed by the LSTM or VAE models, it remains an accurate operational solution for banking experts. This advantage stems from its inherent interpretability and its ability to directly produce meaningful two-dimensional visualizations. In contrast, the competing deep learning models generate high-dimensional latent embeddings that are not readily interpretable and require an additional dimensionality reduction step before they can be effectively visualized.

5.6.2 Conceptual comparison

The proposed SASI framework conceptually differs from conventional RFM-based and VAE-based techniques in their modeling and learning paradigms. VAE-based, static RFM-based and

dynamic RFM-based techniques (whether used in their raw form or enhanced through feature aggregation or LSTM-based encoders) rely on **deterministic** transformations of transactional data into fixed-dimensional feature vectors. These representations are typically constructed through heuristic or learned mappings that summarize the input data, and are subsequently processed by clustering algorithms. In this sense, these approaches are essentially **discriminative** with respect to the clustering objective, as they focus on producing separable feature spaces without explicitly modeling the underlying data-generating process. While LSTM-based encoders and VAEs introduce a learned representation capable of capturing temporal dependencies, they remain deterministic mappings from input sequences to latent vectors and often **lack interpretability**, requiring additional **dimensionality reduction** techniques for visualization.

In contrast, the proposed SASI framework adopts a **probabilistic** and **generative** modeling perspective based on HMMs. Each customer is represented by a HMM λ that explicitly models the joint generative process governing the customer sequential intensity (through state transitions) and sequential amounts (through emission probabilities). This probabilistic formulation enables the estimation of latent behavioral regimes and their temporal dynamics, providing a structured mechanism for handling uncertainty and variability in customer behavior. The meta-data φ and $\overline{\varphi}$ extracted from the learned HMMs yield the **interpretable** pair (x, y) of descriptors, that directly reflect the long-term behavioral patterns of each customer. Unlike dynamic RFM-based and VAE-based representations, which may produce moderate to high dimensional and weakly interpretable trajectories, the SASI framework always maintains a **two-dimensional** and semantically meaningful representation throughout the analysis.

Overall, the key distinction lies in the fact that SASI framework models the stochastic process underlying customer behavior, whereas RFM-based and VAE-based techniques primarily construct descriptive feature spaces, leading to differences in interpretability, robustness, and capacity to capture sequential dynamics. Table 11 summarizes the main findings of the conceptual comparison between the proposed SASI framework and existing deep models.

Table 11: Conceptual comparison between the proposed SASI framework and existing deep models.

Property	**SASI framework**	**Deep models**
Modeling paradigm	Probabilistic	Deterministic
Learning paradigm	Generative	Discriminative
Interpretability	High	Low
Low-dimensional visualization	Direct	Dimensionality reduction

5.7 Main assets

The following key assets can be attributed to the SASI framework proposed in this paper:

1. The proposed SASI framework leverages HMMs to jointly capture both sequential transactional amounts and sequential transactional intensity for each customer, encoded into two interpretable vectors $\overline{\varphi}$ and φ, respectively. To the best of our knowledge, no existing approach provides such a unified HMM-based dual representation.
2. In contrast to existing representations, which typically require dimensionality reduction techniques for visualization due to their high-dimensional embeddings, the proposed SASI framework directly associates each customer with a two-dimensional point (x, y). This enables straightforward visualization in the *Euclidean* two-dimensional plane, which is particularly advantageous for human interpretability.
3. The feature vectors produced by existing related techniques are generally weakly interpretable. Whereas, in the proposed SASI framework, the representation (x, y) has a clear semantic interpretation: x encodes sequential transactional amounts, while y reflects sequential transactional intensity.
4. The proposed framework achieves the best DBI and CHI on the *Nigerian* dataset, and the best SC and DBI on the *Berka* dataset.
5. SASI exhibits satisfactory scalability properties, making it suitable for large-scale customer

data analysis in distributed computing environments.

5.8 Major limitations

Although the proposed SASI framework achieves promising customer segmentation performance, several limitations should be acknowledged.

1. Transaction amounts are normalized with respect to the maximum amount observed for each customer. Although simple and interpretable, this strategy may be sensitive to extreme transactions. More robust normalization methods deserve further investigation.
2. SASI models customer behavior using only transaction amounts and inter-transaction delays. While these variables provide an interpretable behavioral representation, incorporating additional geographic or demographic information could further improve customer characterization.
3. SASI provides a static representation of customer behavior over a fixed observation period. Consequently, it supports customer segmentation and visualization but does not explicitly model behavioral evolution over time.
4. In its current form, SASI can only capture spherical clusters owing the exclusive use of the *K–means* algorithms. However, existing datasets may be naturally composed of non-spherical clusters that cannot actually be accurately captured by SASI, resulting to poor segmentation performances. Exploring alternative clustering strategies and assessing their impact on customer segmentation quality may constitute a relevant direction for future work.

These limitations define the current scope of the proposed framework without affecting the validity of the reported results. They also motivate the future research directions presented in the conclusion, aimed at extending SASI toward more comprehensive and generalizable banking decision-support applications.

5.9 Sensitivity to hyperparameters

The proposed SASI framework is parameterized by a set of user-defined hyperparameters, including T_{min}, T_{max}, N, $\mathcal{M}$, η, r and ε, whose tuning may significantly influence the empirical performance. Consequently, a sensitivity analysis is required to assess the robustness of the framework with respect to these parameters. The findings of this sensitivity analysis are synthesized in Table 12.

6 Conclusion

This paper introduced the SASI framework, an HMM-based approach for customer segmentation that jointly models the sequential dynamics of transaction amounts and transaction intensity. Unlike conventional static and dynamic RFM-based representations or VAE-based embeddings, SASI produces an interpretable two-dimensional customer representation derived from the stationary distributions $\overline{\varphi}$ and φ learned from customer-specific HMMs. The resulting representation enables efficient visualization and clustering using the *K–means* algorithm. Both sequential and parallel Python implementations of the framework, together with the experimental data, are publicly available [5].

Experiments conducted on $1,503$ and $2,039$ customers respectively extracted from the synthetic *Nigerian* and the real-world *Berka* datasets demonstrate the effectiveness of the proposed SASI framework. Indeed, when SASI is compared with the competing approaches, it achieves the second-best SC (0.650635), the best DBI (0.418718), and the best CHI (6578.11) on the *Nigerian* dataset. On the *Berka* dataset, it attains the best SC (0.507139), the best DBI (0.500209), and the second-best CHI (3309.01). These results highlight the ability of SASI to generate compact, well-separated, fully discriminative, and interpretable customer representations suitable for banking decision-support systems. Consequently, although SASI ranks second according to one evaluation metric on each experimental dataset, it remains an accurate operational solution for banking experts. Several directions can be explored in future work:

1. Investigate alternative transaction normalization strategies when using Equation 4, such as percentile-based, z-score, or interquartile-range normalization, to improve robustness to extreme transaction amounts.
2. Extend SASI by incorporating additional behavioral signals (e.g., transaction categories,

Table 12: Sensitivity of the proposed SASI framework to its main hyperparameters.

	Small value	**High value**	**How to tune ?**
T_{min}	Unreliable estimations & Degenerate or overly sparse HMMs	Biased selection & Limited generalizability of the framework	$(T_{min} \geq 3.N^2)$
T_{max}	Discards informative high-activity profiles	Introduces heterogeneity & Time consuming	$(T_{max} \geq 7.N^2)$
N	Underfitting & Inability to capture subtle temporal variations	Sparse transition estimates & Unstable learning	$N^2 \leq \frac{T_{min}}{3}$
$\mathcal{M}$	Masks meaningful variations & Loss of discriminative power in B	Sparse emission probabilities & Reduced robustness	$\mathcal{M} \in [10, 25]$
η	Suboptimal likelihood estimates & Degraded performances	Negligible marginal gain & Time consuming	$\eta \in [300, 500]$
r	Poor stationary distribution	Time consuming	$r \in [500, 1000]$
ε	Reduces flexibility & increase the risk of suboptimal convergence	Distorts the original Markovian structure & Weakens behavioral priors	$\varepsilon \in [10^{-3}, 1]$

payment channels, merchant information, geographical context, or customer demographics) in order to enrich the representational capacity of the HMM-based encoding.

3. Develop a dynamic version of SASI capable of modeling long-term customer behavioral evolution and supporting customer behavioral visualization over time.
4. Investigate alternative clustering algorithms, particularly density-based approaches, to better capture complex customer distributions.
5. Integrate SASI into operational banking decision-support systems for applications such as customer relationship management, fraud detection, credit risk assessment, and personalized financial services.

Statements and Declarations

Code and data availability

The authors declare that the source codes and the experimental data supporting the findings of this study are available online through the following URL: https://github.com/HmmBank/HMM-Based-SASI-Frameworks/.

Materials availability

HTML and PDF visualization files containing the customer segmentation generated by this study are available online through the same following URL: https://github.com/HmmBank/HMM-Based-SASI-Frameworks/.

Authors' contributions

All the authors contributed equally to this work.

Acknowledgements

The authors would like to express their sincere gratitude to Prof. Maurice TCHUENTE for his valuable guidance and insightful advice throughout this work.

Funding

The authors received no specific funding for this work.

Competing interests

The authors declare no competing interests.

Ethics approval and consent to participate

Not applicable.

References

[1] Wedel, M., Kamakura, W.A.: Market Segmentation. Springer, ??? (2000). https://doi.org/10.1007/978-1-4615-4651-1 . http://dx.doi.org/10.1007/978-1-4615-4651-1

[2] Ngai, E.W.T., Xiu, L., Chau, D.C.K.: Application of data mining techniques in customer relationship management: A literature review

and classification. Expert Systems with Applications **36**(2), 2592–2602 (2009) https://doi.org/10.1016/j.eswa.2008.02.021

[3] McKinsey & Company: The future of payments in africa. Industry report, McKinsey Global Institute, New York, NY, USA (2022). https://www.mckinsey.com/industries/financial-services/our-insights/the-future-of-payments-in-africa

[4] Nigeria Inter-Bank Settlement System: Nibss insights: E-payment transactions in nigeria. Technical report, Nigeria Inter-Bank Settlement System, Lagos, Nigeria (2023). https://nibss-plc.com.ng/e-payment-transactions-in-nigeria-hit-monthly-all-time-high-of-33-2-trillion-in-august-2022/

[5] Černevičienė, J., Kabašinskas, A.: Explainable artificial intelligence (xai) in finance: a systematic literature review. Artificial Intelligence Review **57**(8) (2024) https://doi.org/10.1007/s10462-024-10854-8

[6] Doğan, O., Ayçin, E., Bulut, Z.A.: Customer segmentation by using rfm model and clustering methods: A case study in retail industry. International Journal of Contemporary Economics and Administrative Sciences **8**(1), 1–19 (2018)

[7] Aliyev, M., Ahmadov, E., Gadirli, H., Mammadova, A., Alasgarov, E.: Segmenting bank customers via rfm model and unsupervised machine learning. arXiv preprint arXiv:2008.08662 (2020) arXiv:2008.08662

[8] Alves Gomes, M., Meisen, T.: A review on customer segmentation methods for personalized customer targeting in e-commerce use cases. Information Systems and e-Business Management **21**(3), 527–570 (2023) https://doi.org/10.1007/s10257-023-00640-4

[9] Mancisidor, R.A., Kampffmeyer, M., Aas, K., Jenssen, R.: Segment-based credit scoring using latent clusters in the variational autoencoder. arXiv preprint arXiv:1806.02538 (2018)

[10] Mancisidor, R.A., Kampffmeyer, M., Aas, K., Jenssen, R.: Learning latent representations of bank customers with the variational autoencoder. Expert Systems with Applications **164**, 114020 (2021) https://doi.org/10.1016/j.eswa.2020.114020

[11] Barkhordar, E., Shirali-Shahreza, M.H., Sadeghi, H.R.: Clustering of bank customers using lstm-based encoder–decoder and dynamic time warping. arXiv preprint arXiv:2110.11769 (2021) arXiv:2110.11769 [cs.LG]

[12] Balbal, K.F., Birant, D.: Rfm-net: A convolutional neural network for customer segment classification. Applied Sciences **16**(5), 2223 (2026) https://doi.org/10.3390/app16052223

[13] Sembiring Brahmana, R.W., Mohammed, F.A., Chairuang, K.: Customer segmentation based on rfm model using k-means, k-medoids, and dbscan methods. Lontar Komputer : Jurnal Ilmiah Teknologi Informasi **11**(1), 32 (2020) https://doi.org/10.24843/lkjiti.2020.v11.i01.p04

[14] Rajesh, M.V., Chintalapudi, S.R., Prasad, M.H.M.K.: A comparative analysis of rfm-based customer segmentation with k-means and birch clustering techniques. In: Proceedings of the International Conference on Computational Innovations and Emerging Trends (ICCIET 2024), pp. 977–989. Atlantis Press International BV, ??? (2024). https://doi.org/10.2991/978-94-6463-471-6_94

[15] Rusly, F., Purba, R., Pasha, M.F.: New approach: Customer segmentation using rfm model and demand classification. Jurnal Teknologi dan Manajemen Informatika **11**(2), 165–176 (2025) https://doi.org/10.26905/jtmi.v11i2.16208

[16] Grigorova, I., Efremov, A., Karamfilov, A.: An automated machine learning framework for interpretable customer segmentation in financial services. International Journal of Financial Studies **13**(4), 243 (2025) https://doi.org/10.3390/ijfs13040243

[17] Filho, A.L.C.V., Lima, L., Kleina, M.: A

graph-based approach to customer segmentation using the rfm model (2025) https://doi.org/10.48550/ARXIV.2505.08136

[18] Abbasimehr, H., Shabani, M.: A new methodology for customer behavior analysis using time series clustering: A case study on a bank's customers. Kybernetes **50**(2), 221–242 (2019) https://doi.org/10.1108/k-09-2018-0506

[19] Abbasimehr, H., Shabani, M.: A new framework for predicting customer behavior in terms of rfm by considering the temporal aspect based on time series techniques. Journal of Ambient Intelligence and Humanized Computing **12**(1), 515–531 (2020) https://doi.org/10.1007/s12652-020-02015-w

[20] Abbasimehr, H., Bahrini, A.: An analytical framework based on the recency, frequency, and monetary model and time series clustering techniques for dynamic segmentation. Expert Systems with Applications **192**, 116373 (2022) https://doi.org/10.1016/j.eswa.2021.116373

[21] Wang, S., Sun, L., Yu, Y.: A dynamic customer segmentation approach by combining lrfms and multivariate time series clustering. Scientific Reports **14**(1) (2024) https://doi.org/10.1038/s41598-024-68621-2

[22] Peker, S., Kocyigit, A., Eren, P.E.: Lrfmp model for customer segmentation in the grocery retail industry: a case study. Marketing Intelligence and Planning **35**(4), 544–559 (2017) https://doi.org/10.1108/mip-11-2016-0210

[23] Allegue, S., Abdellatif, T., Bannour, K.: Rfmc: a spending-category segmentation. In: 2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 165–170. IEEE, ??? (2020). https://doi.org/10.1109/wetice49692.2020.00040 . http://dx.doi.org/10.1109/WETICE49692.2020.00040

[24] Zhou, J., Wei, J., Xu, B.: Customer segmentation by web content mining. Journal of Retailing and Consumer Services **61**, 102588 (2021) https://doi.org/10.1016/j.jretconser.2021.102588

[25] Ho, T., Nguyen, S., Nguyen, H., Nguyen, N., Man, D.-S., Le, T.-G.: An extended rfm model for customer behaviour and demographic analysis in retail industry. Business Systems Research Journal **14**(1), 26–53 (2023) https://doi.org/10.2478/bsrj-2023-0002

[26] Martiansah, R., Monalisa, S., Muttakin, F., Fronita, M.: Customer segmentation analysis through rfm-d model and k-means algorithm. Jurnal Sistem Cerdas **8**(1), 1–11 (2025) https://doi.org/10.37396/jsc.v8i1.504

[27] Pandey, A.K., Goyal, A., Sikka, N.: Re-rfme: Real-estate rfme model for customer segmentation. arXiv preprint arXiv:2404.17177 (2024) arXiv:2404.17177 [cs.LG]

[28] Lu, Z., Kannan, P.K.: Ai for customer journeys: A transformer approach. Journal of Marketing Research **63**(1), 1–26 (2025) https://doi.org/10.1177/00222437251347268

[29] Shumovskaia, V., Fedyanin, K., Sukharev, I., Berestnev, D., Panov, M.: Linking bank clients using graph neural networks powered by rich transactional data. International Journal of Data Science and Analytics **12**(2), 135–145 (2021) https://doi.org/10.1007/s41060-021-00247-3

[30] Pal, A., Mittal, J., Sharma, A., Annupriya: Network-aware customer churn prediction in banking using graph machine learning techniques, 238–249 (2026) https://doi.org/10.1007/978-3-032-29237-7_21

[31] Danielsson, O., Hernberg, J.: Uncovering customer segments from spending and demographic data with hdbscan: Defining customer groups from spending data of bank customers. Master's thesis, KTH Royal Institute of Technology, Stockholm, Sweden (2025). Independent thesis, advanced level (30 ECTS)

[32] Ching, W.-K.: Hidden markov models and their applications to customer relationship management. IMA Journal of Management Mathematics **15**(1), 13–24 (2004) https://doi.org/10.1093/imaman/15.1.13

[33] Netzer, O., Lattin, J.M., Srinivasan, V.: A hidden markov model of customer relationship dynamics. Marketing Science **27**(2), 185–204 (2008) https://doi.org/10.1287/mksc.1070.0294

[34] Jandera, A., Skovranek, T.: Customer behaviour hidden markov model. Mathematics **10**(8), 1230 (2022) https://doi.org/10.3390/math10081230

[35] Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989) https://doi.org/10.1109/5.18626

[36] Lange, K., Zhou, H.: A legacy of em algorithms. International Statistical Review **90**(S1) (2022) https://doi.org/10.1111/insr.12526

[37] Kang, J.: STAT 243: Stochastic Process. https://bookdown.org/jkang37/stochastic-process-lecture-notes/lecture09.html (2021)

[38] Tsagmo Denkeng, A.J., Mohammad Mourad, A., Iloga, S., Mballa Mba, R.M., Baazaoui, H., Djotio Ndié, T., Romain, O.: Efficient characterization of the human balance using hmms. IEEE Access **13**, 183456–183479 (2025) https://doi.org/10.1109/access.2025.3622375

[39] Chiasseup Keudjeu, J., Kengni Ngangmo, O., Iloga, S., Bouetou Bouetou, T.: Hybrid movie recommendation based on recent user sequential preferences and sequential profiles using hmms. IEEE Access **13**, 192330–192357 (2025) https://doi.org/10.1109/access.2025.3630634

[40] Yang, F., Balakrishnan, S., Wainwright, M.J.: Statistical and computational guarantees for the baum–welch algorithm. Journal of Machine Learning Research **18**(21), 4528–4580 (2017)

[41] Rau, N., Lücke, J., Hartmann, A.K.: Phase transition for parameter learning of hidden markov models. Physical Review E **104**(4) (2021) https://doi.org/10.1103/physreve.104.044105

[42] Electric Sheep Africa: Nigerian Retail Banking Transactions Dataset. https://huggingface.co/datasets/electricsheepafrica/nigerian-retail-transactions

[43] Berka, P.: PKDD'99 Discovery Challenge Dataset (Berka Dataset). Proceedings of the PKDD'99 Discovery Challenge. Available through the UCI Machine Learning Repository and multiple public mirrors (1999)

[44] Frank, E., Hall, M.A., Witten, I.H., Reutemann, P., Trigg, L.: WEKA: The Waikato Environment for Knowledge Analysis (Version 3.9.3). https://ml.cms.waikato.ac.nz/weka/, Hamilton, New Zealand. Open-source machine learning software developed at the University of Waikato. GNU General Public License (2018)

[45] Dalcin, L., Fang, Y.-L.L.: mpi4py: Status update after 12 years of development. Computing in Science and Engineering **23**(4), 47–54 (2021) https://doi.org/10.1109/mcse.2021.3083216

[46] Rogowski, M., Aseeri, S., Keyes, D., Dalcin, L.: mpi4py.futures: Mpi-based asynchronous task execution for python. IEEE Transactions on Parallel and Distributed Systems **34**(2), 611–622 (2023) https://doi.org/10.1109/tpds.2022.3225481