Proceedings of the 12th RSI International Conference on Robotics and Mechatronics (ICRoM 2024), Dec. 17-19, 2024, Tehran, Iran

# Bridging Complexity and Interpretability: A Two-Phase Clustering Framework

D. A. Nejad*, A. Mehrabi*, A. Rezaei *, A. Jahani*, S. A. Khalilpour*, H. Kh. Seyedi*, and H. D. Taghirad*
* Applied Robotics and AI Solutions (ARAS), K. N. Toosi University of Technology, Tehran, Iran.
Email: *taghirad@kntu.ac.ir

*Abstract*—**Unsupervised learning has consistently been a challenging aspect of machine learning, particularly in robotic tasks such as autonomous navigation, object grasping, and anomaly detection. These tasks require the system to identify patterns in data without any labeled examples, making it difficult to achieve reliable and accurate outcomes. In medical applications, such as tumor detection and segmentation, the difficulties of unsupervised learning become even more noticeable. Without labeled data to guide the models, discovering meaningful patterns often leads to poor interpretability, limiting the effectiveness of these systems in critical medical decision-making processes. In these contexts, transparency and understanding are essential, especially for safety-critical tasks. Recognizing this need, we introduce a novel two-phase clustering approach that combines latent space transformation using an autoencoder with weighted feature augmentation and post-clustering in the original feature space, thereby improving explainability by making the clusters more interpretable and highlighting key features for a better understanding of the results. Building on this, our method, applied to a real-world dataset characterized by complex human behaviors developed by our group, demonstrates superior performance with improved silhouette scores and comprehensive analysis, effectively capturing complex customer behaviors while enhancing interpretability and providing key insights for data-driven decisions.**

*Index Terms*—**Unsupervised Learning, Clustering, Autoencoders, Explainable AI, Human Behavior Analysis**

## I. Introduction

In recent years, unsupervised learning has become a fundamental approach for enhancing robotic capabilities. This allows robots to autonomously discover data patterns and structures without needing labeled examples, improving their adaptability and efficiency in dynamic environments [1]. However, as these autonomous systems grow more complex, the demand for explainability in their decision-making processes has increased, especially in safety-critical applications [2].

Explainable unsupervised learning allows robots to learn from unlabeled data while providing human-understandable insights into clustering, feature extraction, and anomaly detection [3]. Recent advancements, such as saliency maps, allow robots to highlight key environmental features influencing their actions, making their behavior more transparent and interpretable [4]. In medical applications like MRI segmentation, unsupervised learning helps detect and analyze tumors, with interpretability crucial for accurate diagnosis and treatment planning [5]. Furthermore, by integrating natural language explanations, robots can also articulate their decisions, enhancing trust and collaboration in human-robot interactions [6]. These developments indicate that the synergy between unsupervised learning and explainability not only enhances robotic autonomy but also promotes greater acceptance and trust in robotics and medical applications across various domains.

KMeans has gained popularity in robotics for tasks such as object detection, feature extraction, and localization due to its efficiency and simplicity. Elango et al. [7] applied KMeans to balance workloads and minimize travel distance in multi-robot task allocation. However, the approach provides limited insight into how clusters are formed, making task distribution difficult to interpret. Additionally, in exploration scenarios, KMeans-based map segmentation lacks transparency in determining how clusters of unknown areas are identified, reducing explainability in dynamic environments [8].

KMeans is widely used beyond robotics, especially in medical segmentation tasks. Alashwal et al. [9] applied it to Alzheimer's data, clustering MRI and cognitive test results to identify patients at higher risk of progressing from mild cognitive impairment. However, KMeans' lack of complex pattern recognition and sensitivity to outliers limit its effectiveness. Zhou et al. [10] combined KMeans with deep autoencoders to classify patients based on disease progression, significantly enhancing diagnostic accuracy. However, KMeans' inherent lack of interpretability presents a major challenge in the medical field, where understanding the rationale behind patient clustering is crucial for clinical decision-making. Additionally, its failure to capture the latent structure in complex datasets often leads to less precise results. Lian et al. [11] applied KMeans to segment MRI images, improving tissue identification. Nonetheless, its sensitivity to outliers and its inability to model the latent representations in medical images remain significant obstacles to achieving more reliable outcomes.

Another application of KMeans clustering can be found in the market business, particularly in customer segmentation. By utilizing KMeans, businesses can uncover hidden patterns and segment customers based on their purchasing behaviors and interactions. This enables companies to create more tailored marketing strategies, optimize resource distribution, and improve customer engagement, all of which contribute to higher profitability and efficiency [12], [13].

As customer segmentation has evolved, KMeans and models like RFM (Recency, Frequency, Monetary) have become essential for analyzing large customer datasets [14]. However, RFM's limited feature set often misses subtle customer behaviors [15]. Advanced techniques like autoencoders help

979-8-3315-2973-4/24/$31.00 ©2024 IEEE

model complex, non-linear patterns in data. For example, Nguyen (2021) introduced deep embedding clustering, combining neural networks with probabilistic clustering to uncover latent segments. While these methods improve segmentation accuracy, the interpretability of deep learning models remains a challenge, making it harder for businesses to fully understand customer behavior drivers [16].

In this paper, we propose a novel two-phase clustering approach that leverages both latent space representations and weighted post-clustering in the original feature space. This methodology is designed to balance the global structure captured in the latent space with the detailed features in the original space, providing a flexible and robust solution for unsupervised tasks. One of the key contributions of our approach is its versatility, making it applicable to a wide range of feature-based unsupervised learning tasks across various domains. Not only does it effectively capture the latent representations of the data through autoencoders, but it also ensures complete interpretability of the resulting clusters. This interpretability allows for valuable insights that can be readily understood and utilized for data-driven decision-making. The effectiveness of our approach is demonstrated through a comprehensive analysis of the transactional dataset developed by our team from a national chain store in the food industry, highlighting the benefits of integrating autoencoders and clustering for deeper insights into human behavior. The enhanced interpretability of this method makes it particularly suited for applications where understanding the underlying data patterns is critical.

The remainder of this paper is organized as follows: Section II details the two-phase clustering approach, including latent space transformation, initial clustering, weighted feature augmentation, and post-clustering. Section III presents the experimental setup, results, and analysis, covering dataset preparation, feature construction, clustering performance, and segment interpretation. Finally, Section IV concludes with a summary of contributions.

## II. Methodology

This section outlines the proposed two-phase clustering approach, which leverages both latent space representations and weighted post-clustering in the original feature space. The methodology includes latent space transformation using an autoencoder, initial clustering, weighted feature augmentation, and post-clustering.

### A. Autoencoder

An autoencoder [17] is a type of artificial neural network primarily used for unsupervised learning tasks such as dimensionality reduction and feature extraction. It consists of two main components: an encoder and a decoder. The encoder compresses the input data into a lower-dimensional latent representation, while the decoder attempts to reconstruct the original input from this compressed form. By minimizing the difference between the input and its reconstruction, autoencoders effectively learn compact and informative features, making them particularly useful for reducing the complexity of high-dimensional data while retaining essential information.

In this work, we employ a fully connected autoencoder architecture to reduce the dimensionality of our dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where each data point $\mathbf{x}_i \in \mathbb{R}^d$. The autoencoder transforms the data into a 12-dimensional latent space $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n\}$, with each $\mathbf{z}_i$ representing the learned compressed feature vector. This transformation is denoted as $\mathbf{z}_i = f_{\text{encoder}}(\mathbf{x}_i)$, where $f_{\text{encoder}}$ refers to the encoder function. Additionally, the autoencoder refines these latent representations, ensuring that the learned features capture the most significant and distinguishing characteristics of the original data, which is crucial for subsequent clustering and analysis tasks.

**Autoencoder Architecture:**

- **Encoder:** A series of fully connected dense layers with units (32, 64, 128, 64, 32), each followed by ReLU activation, and ending with a 12-dimensional bottleneck layer representing the compressed latent space.
- **Decoder:** Mirrors the encoder with fully connected dense layers (32, 64, 128, 64, 32 units), tasked with reconstructing the input from the 12-dimensional latent space.

The autoencoder is trained by minimizing the reconstruction error between the input $\mathbf{x}_i$ and its output $\hat{\mathbf{x}}_i$ using the Mean Squared Error (MSE) loss function [18], which evaluates the quality of the reconstruction as follows:

$$\text{MSE}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{d} \sum_{j=1}^{d} (\mathbf{x}_{ij} - \hat{\mathbf{x}}_{ij})^2 \tag{1}$$

The optimization is performed using the Adam optimizer [19].

### B. Initial Clustering

Clustering is first performed in the latent space $\mathbf{Z}$ using KMeans, which partitions the data into $k$ clusters by minimizing the within-cluster variance, a method widely recognized for its effectiveness in unsupervised learning tasks [20]:

$$\min_{\{\mathbf{C}_j\}} \sum_{i=1}^{n} \min_{j} \|\mathbf{z}_i - \mathbf{C}_j\|^2 \tag{2}$$

This process generates initial cluster labels $\mathbf{L} = \{l_1, l_2, \ldots, l_n\}$.

### C. Post-Clustering

To integrate the clustering information from the latent space into the original data space, each data point $\mathbf{x}_i$ is augmented with a one-hot encoded vector $\mathbf{e}_{l_i}$ representing the latent space cluster label, weighted by a factor $\alpha$. This technique is inspired by methods used in feature engineering to emphasize the importance of certain features in clustering [21]:

$$\mathbf{X}'_i = [\mathbf{x}_i, \alpha \mathbf{e}_{l_i}] \tag{3}$$

KMeans clustering is then applied to the augmented dataset $\mathbf{X}' = \{\mathbf{X}'_1, \mathbf{X}'_2, \ldots, \mathbf{X}'_n\}$ with the objective:

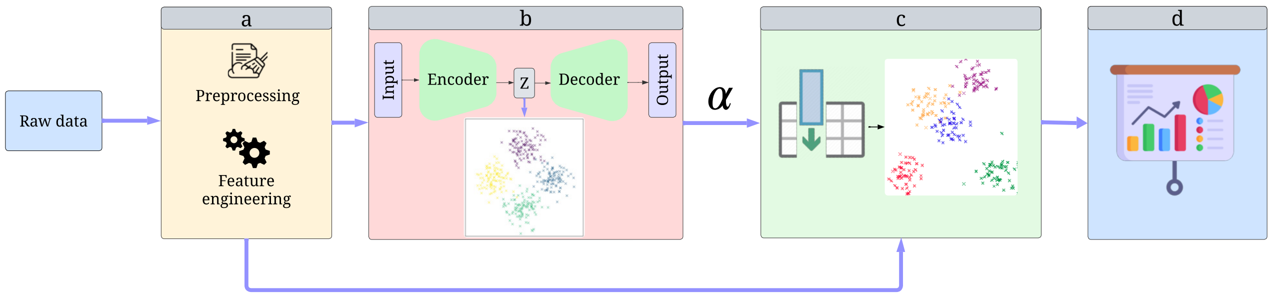


Fig. 1. Model's workflow

$$\min_{\{\mathbf{C}'_j\}} \sum_{i=1}^{n} \min_j \|\mathbf{X}'_i - \mathbf{C}'_j\|^2 \tag{4}$$

This step refines the clusters by balancing the global structure captured in the latent space with the detailed features in the original space, guided by the weighted latent labels. The weight $\alpha$ controls the influence of the latent space clustering, allowing for a flexible and robust clustering solution.

The overall structure of the workflow is depicted in Figure 1. As outlined, the process begins with step (a), where preprocessing and feature engineering of the raw data are conducted—details of which will be elaborated in the subsequent chapter. Following this, in step (b), the proposed model is applied, where the data is processed through an autoencoder to obtain latent representations, leading to the formation of initial clusters. Subsequently, in step (c), these weighted initial clusters are integrated with the features generated in step (a), upon which post-clustering is performed. Lastly, in step (d), the analysis of clusters and the development of customer profiles are carried out using explainability methods, which will be thoroughly discussed in the following chapter.

## III. Implementation Results

This section presents a detailed analysis of the implementations carried out, the results obtained, and the implications of these findings. The analysis is organized into three primary areas: dataset preparation and feature construction, clustering analysis and model comparison, and the interpretability of the resulting customer segments.

### A. Dataset and Feature Construction

As we aim to demonstrate the interpretability and capability of our model in uncovering underlying trends, we implemented it on a transactional dataset that reflects complex human behavior, which is inherently difficult to capture. The dataset, sourced from a national food chain store, covers transactional data from 2022 to 2024. It includes transaction dates, customer IDs, article IDs (representing purchased items), the quantity of articles per transaction, the branch where the purchase occurred, and the payment method. This real-world dataset provides a rich foundation for testing our approach's ability to reveal hidden patterns and deliver meaningful insights into customer interactions. A short version of the dataset is available in the github repository linked on the page[1].

To prepare the data for analysis, we implemented a comprehensive preprocessing strategy. Missing values in key fields like customer IDs and article IDs were imputed to maintain data integrity and prevent bias. Anomalous or invalid entries, such as incorrect user or article IDs, were identified and corrected, ensuring that the dataset accurately reflected typical customer behavior. Outliers in transaction amounts and purchase frequencies were identified and removed using the isolation forest method, which is effective in detecting anomalies by isolating observations in the dataset [22]. This approach ensures the analysis focuses on standard customer behaviors. After preprocessing, the dataset consisted of 10,410 unique customers.

Following data preprocessing, feature engineering was conducted to transform the raw transactional data into a structured format suitable for clustering. The 22 derived behavioral features were categorized into three main types: (1) numerical features, such as the number of transactions, total transaction value, and days since the last purchase; (2) categorical features, including the most frequent day of the week for purchases and the most frequent payment methods; and (3) mean or ratio-based features, such as spending trends, repeat purchase ratios, and average intervals between purchases.

These features were carefully chosen to capture key customer behaviors, creating a strong foundation for clustering. By transforming raw transactional data into these informative features, we enhance the ability to differentiate customer segments, enabling more effective segmentation and targeted marketing strategies, as supported by existing studies in customer segmentation.

### B. Clustering Analysis and Comparative Study

In this section, we analyze the clustering results and compare the effectiveness of the proposed model with KMeans and Autoencoder-based clustering without stacking. The analysis includes determining the optimal number of clusters, evaluating the clustering performance using metrics such as the elbow

[1] https://github.com/aras-labs/ARAS-RS

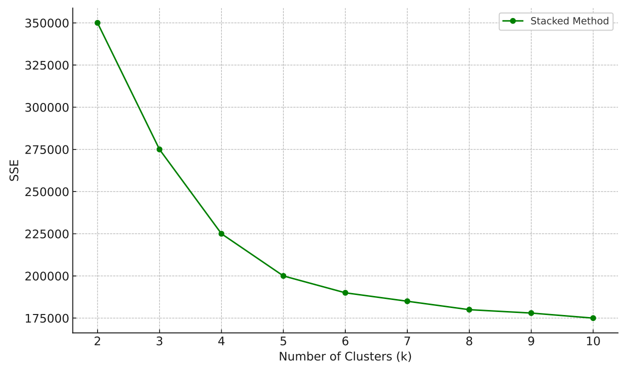


Fig. 2. Elbow method for stacked method

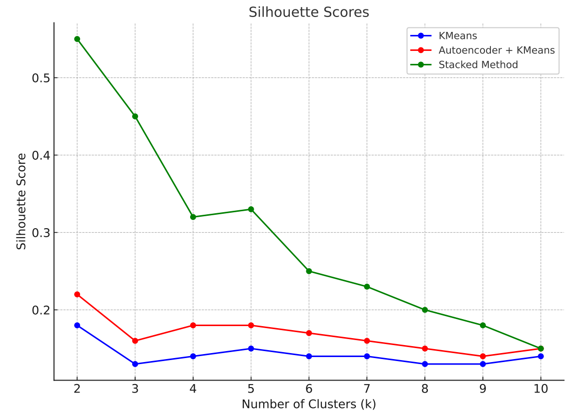


Fig. 3. Silhouette scores for different clustering methods across varying numbers of clusters (k).

method and silhouette score, and comparing the clustering results and errors across different models.

*1) Determining the Optimal Number of Clusters:* To identify the suitable number of clusters, we applied both the elbow method [23] and the silhouette score [24] for varying numbers of clusters. As shown in Figure 2, the elbow method involves plotting the sum of squared errors (SSE) against the number of clusters. The optimal number of clusters is typically identified at the "elbow point," where the SSE curve begins to flatten, indicating diminishing returns from adding more clusters. In our analysis, the elbow point, as shown in Figure 2, suggests that the optimal number of clusters falls within the range of 5 to 6. Furthermore, the silhouette score, which quantifies how well an object is matched to its own cluster relative to other clusters Figure 3, supports the selection of 4 or 5 clusters. Considering both methods, we selected 5 as the optimal number of clusters for our model, balancing cluster cohesion and separation.

*2) Clustering Results and Error Analysis:* To evaluate the performance of the proposed stacked clustering model, we compared it against two baseline methods KMeans and Autoencoder without stacking. As depicted in the Figure 3, the stacked method consistently achieves the highest silhouette scores across different numbers of clusters (k). Starting with a score of over 0.5 for two clusters, the silhouette score gradually decreases as the number of clusters increases, yet it remains significantly higher than both the Autoencoder+KMeans and KMeans methods. This demonstrates that the stacked model forms more distinct and well-separated clusters.

In contrast, the Autoencoder+KMeans method exhibits moderate silhouette scores, with a peak at k = 2 but significantly lower scores than the stacked method for higher values of k. The KMeans method shows the lowest silhouette scores, remaining consistently below 0.2 across different cluster numbers, indicating poorer cluster formation and higher overlap between clusters.

This analysis confirms that the proposed stacked method offers superior clustering performance, effectively balancing distinct and well-defined clusters while capturing the complexity of customer behavior. The results demonstrate that this stacked approach provides more robust and cohesive clusters, outperforming traditional KMeans or Autoencoder-based methods in identifying distinct customer groups.

We analyzed the resulting clusters using t-SNE (t-Distributed Stochastic Neighbor Embedding) visualizations, as shown in Figure 4. This figure displays three t-SNE plots: one for the stacked model (a), one for the Autoencoder-based method (b), and one for the KMeans method (c).

The stacked model demonstrates more distinct and well-separated clusters, with minimal overlap between different groups, suggesting that this approach better captures the inherent structure of the data. In contrast, the Autoencoder-based method without stacking (b) shows a higher degree of overlap between clusters, indicating less cohesive cluster formation. Similarly, the KMeans method (c) results in even more overlapping clusters, making it difficult to distinguish between the different customer segments.

### C. *Interpretability of Customer Segments*

Understanding the drivers behind each customer segment is crucial for effective marketing strategies. In this study, a combination of statistical analyses and visualization techniques was employed to enhance segment interpretability. Statistical measures such as mean, median, standard deviation, and correlation coefficients provided a quantitative foundation for distinguishing customer behaviors, while visual tools like radar charts, box plots, and histograms offered multi-dimensional insights into spending habits, transaction frequency, and purchasing patterns across clusters.

Radar charts were instrumental in comparing key features across segments, whereas box plots effectively summarized the distribution of essential attributes such as transaction value and frequency, highlighting variability and consistency within each group. Histograms further clarified the frequency distributions, aiding in identifying common patterns and outliers in customer behavior.

By integrating these methods, a comprehensive understanding of each customer segment was achieved, enabling busi-

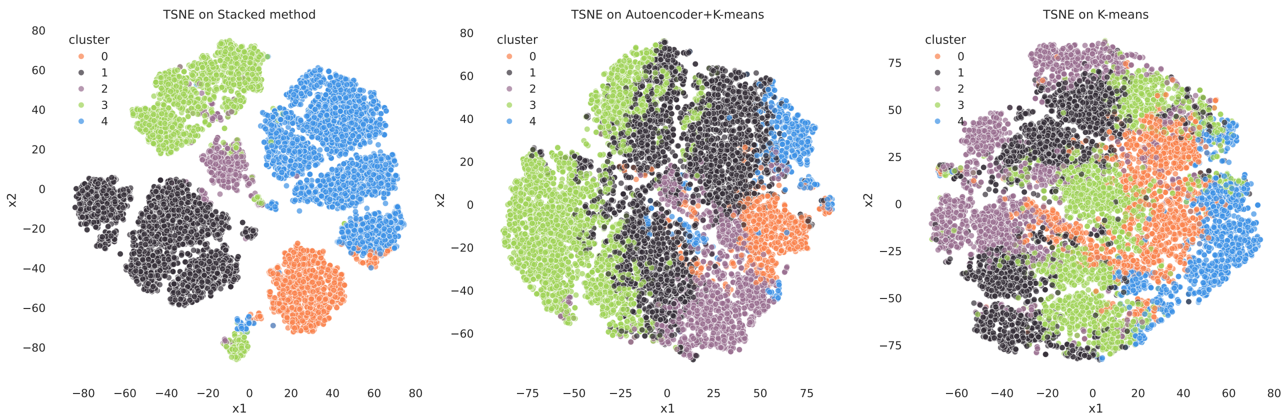


Fig. 4. t-SNE visualization of clustering results: (a) Stacked clustering method, (b) Autoencoder method, (c) KMeans method.

nesses to craft more targeted marketing strategies aligned with specific customer behaviors and preferences [25], [26]. The analysis revealed distinct behavioral patterns and engagement levels, which are visualized through boxplots in Figure 5, providing valuable insights for informed decision-making.

***Cluster 0 (At-Risk, High-Value Buyers):*** Cluster 0 is characterized by customers who, despite engaging in relatively few transactions, contribute significantly to total revenue. These customers are considered to be old customers, as it has been a long period since their first transaction. While their high average transaction value suggests a preference for larger purchases, the small volume of transactions and the extended Max Time Without Purchase indicate that they may be at risk of leaving or have already disengaged from the brand. This cluster also shows high variability in monthly spending, highlighting fluctuating purchasing behavior over time.

***Cluster 1 (Newer, Consistent Spenders):*** Cluster 1 includes customers who are relatively new, as indicated by the short period since their first purchase. These customers are highly engaged in terms of transaction frequency and contribute significantly to overall spending. They make frequent, consistent transactions, and the Max Time Without Purchase is the shortest in this cluster, indicating regular engagement. Additionally, these customers purchase from a variety of product categories, and their spending patterns remain stable with moderate variability.

***Cluster 2 (Low-Value, Transient Buyers):*** Cluster 2 represents customers with the lowest average transaction value and the fewest number of purchases. Their spending is generally balanced across transactions, but their purchasing behavior suggests they are transient customers who are unlikely to make repeat purchases. These customers tend to make their purchases in the middle of the week, which further supports their transient nature. While their Max Time Without Purchase is moderate, the low variability in their monthly spending indicates a steady but minimal engagement with the brand.

***Cluster 3 (Loyal, High-Value Frequent Buyers):*** Cluster 3 represents long-standing and loyal customers, as indicated by the significant time since their first purchase and the short time since their last purchase. These customers are highly engaged with the brand, demonstrating continuous purchasing behavior. They contribute the most to total revenue, with a high volume of purchases that includes a variety of specific products. Their purchases are generally made on the weekends, and their monthly spending is stable, with low variability.

***Cluster 4 (Low-Commitment, Inconsistent Buyers):*** Cluster 4 includes customers who are similar to Cluster 2 in terms of characteristics, but with the distinction that most of their purchases are made on the weekend. Furthermore, this cluster has a larger number of customers compared to Cluster 2, despite their overall low engagement with the brand. This suggests that the brand may not fully resonate with their preferences, or they may not feel entirely satisfied with their experience. They are likely to be normal customers who do not have a strong commitment to the brand, resulting in a low number of repeat purchases and average purchase volume. Additionally, this cluster exhibits high variability in monthly spending, indicating inconsistent purchasing behavior.

## IV. Conclusions

In this study, we proposed a two-phase clustering approach that integrates latent space transformation using an autoencoder with weighted feature augmentation and post-clustering in the original feature space. This methodology demonstrated significant advantages in capturing complex, non-linear patterns in customer behavior, leading to more accurate and meaningful customer segmentation compared to traditional methods. Our implementation results showed that the stacked clustering approach outperformed standard KMeans and autoencoder-based clustering techniques in terms of clustering quality, as evidenced by higher silhouette scores and more distinct, well-separated clusters in t-SNE visualizations.

The interpretability of the resulting clusters was enhanced through comprehensive statistical and visual analyses, revealing distinct customer patterns, such as high-value loyal customers, at-risk buyers, and low-engagement segments. These insights support targeted marketing strategies and personalized engagement to boost customer satisfaction and retention. By balancing latent space representations with interpretability, our adaptable approach provides a versatile framework for feature-based unsupervised learning tasks. This framework enables insightful analysis and decision-making, ensuring its applicability to a wide range of scenarios across domains like healthcare, finance, and industrial processes, where both accuracy and clear, interpretable results are essential.

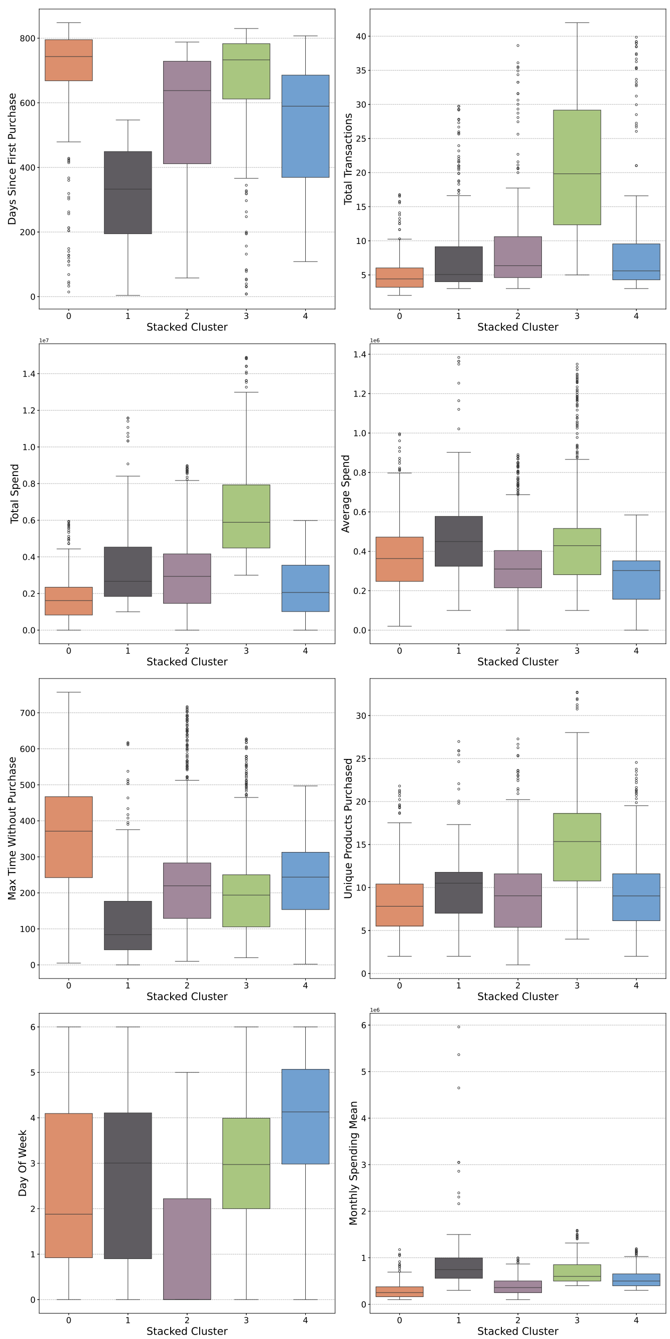


Fig. 5. Box plots of important features

## References


1 C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

2 F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

3 A. Chatzimparmpas, R. M. Martins, and A. Kerren, "A survey of surveys on the use of visualization for interpreting machine learning models," *Information Visualization*, vol. 19, no. 3, pp. 207-233, 2020.

4 Q. Zhang and S. C. Zhu, "Visual interpretability for deep learning: a survey," *Frontiers of Information Technology Electronic Engineering*, vol. 19, no. 1, pp. 27-39, 2018.

5 G. Litjens, T. Kooi, B. E. Bejnordi, A. A. Setio, F. Ciompi, M. Ghafoorian, ... C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60-88, 2017.

6 U. Ehsan, B. Harrison, L. Chan, and M. O. Riedl, "Automated rationale generation: A method for explainable AI and its effects on human perceptions," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 2019, pp. 263-274.

7 M. Elango, S. Nachiappan, and M. K. Tiwari, "Balancing task allocation in multi-robot systems using K-means clustering and auction based mechanisms," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6486–6491, 2011.

8 L. Goodwin and S. Nokleby, "A K-Means Clustering Approach to Segmentation of Maps for Task Allocation in Multi-robot Systems Exploration of Unknown Environments," *Proceedings of the 2022 USCToMM Symposium on Mechanical Systems and Robotics*, vol. 118, pp. 235–245, 2022.

9 H. Alashwal, M. El Halaby, J. J. Crouse, A. Abdalla, and A. A. Moustafa, "The Application of Unsupervised Clustering Methods to Alzheimer's Disease," *Frontiers in Computational Neuroscience*, vol. 13, p. 31, 2019. doi: 10.3389/fncom.2019.00031.

10 Y. Zhou, L. Wang, and M. Huang, "Joint Optimization of Autoencoder for Clustering and Embedding,", Machine Learning, Springer, 2019.

11 C. Lian, X. Wu, and H. Huang, "Medical Imaging and Informatics: Computational Models and Applications,", 2015, pp. 87-95.

12 S. Gupta and V. Zeithaml, "Customer Metrics and Their Impact on Financial Performance," *Marketing Science*, vol. 25, no. 6, pp. 718–739, 2006.

13 D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, John Wiley & Sons, 2015.

14 I. J. Chen and K. Popovich, "Understanding Customer Relationship Management (CRM): People, Process, and Technology," *Business Process Management Journal*, vol. 9, no. 5, pp. 672–688, 2003.

15 P. Anitha and M. M. Patil, "RFM model for customer purchase behavior using K-Means algorithm," *Journal of King Saud University*, 2019.

16 H. H. Nguyen, "Deep Learning for Customer Segmentation in E-Commerce," *Journal of Marketing Analytics*, vol. 7, no. 4, pp. 155-165, 2021.

17 G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, July 2006.

18 I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

19 D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

20 X. Jin and J. Han, "K-Means Clustering," *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer, 2011, pp. 563-564.

21 I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

22 F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *2008 Eighth IEEE International Conference on Data Mining*, pp. 413-422, 2008.

23 D. J. Ketchen and C. L. Shook, "The application of cluster analysis in strategic management research: An analysis and critique," *Strategic Management Journal*, vol. 17, no. 6, pp. 441–458, 1996.

24 P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

25 D. D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical Statistics with Applications*, 7th ed., Cengage Learning, 2014.

26 I. Spence and R. F. Garrison, "A remarkable scatterplot," *The American Statistician*, vol. 47, no. 1, pp. 12–19, 1993.