Advances in Consumer Research
ISSN (Print): 0098-9258
ISSN(Online): 3079-1766

Vol. 2, Issue 2 (2025) https://acr-journal.com/

Mastering Retail Segmentation: Exploring Clustering Algorithms for Precise Market Insights

Jolly Masih[1], Dinesh Kumar Yadav[2], Suresh Sharma[3], Neha Saini[*4]

[1]Associate Professor, BML Munjal University, Gurugram, Haryana, India
[2]Dev Bhoomi Uttarakhand University, Dehradun, India
[3]Assistant Professor, CCS National Institute of Agriculture Marketing, Jaipur, Rajasthan, India
[4*]Educator in Agri Business and Capacity Building Facilator in Food Business, Gurugram, Haryana, India
***Corresponding Author:**
Email ID: nehamanage09@gmail.com

Cite this paper as: Jolly Masih, Dinesh Kumar Yadav, Suresh Sharma, Neha Saini, (2025) Mastering Retail Segmentation: Exploring Clustering Algorithms for Precise Market Insights. *Advances in Consumer Research*, 2 (2), 252-267.

KEYWORDS

Customer Segmentation, Clustering Algorithms, Evofusion Model, Dimensionality Reduction, Deep Learning.

ABSTRACT

This study aimed to transform customer segmentation in the retail sector by systematically evaluating and contrasting various clustering algorithms to generate deeper market insights. The preprocessing phase integrates key data preparation techniques, including mean imputation to address missing values, Z-Score Standardization for normalization, and One-Hot Encoding to manage categorical data. To counter data imbalance, the Adaptive Synthetic Sampling (ADASYN) technique was employed. Dimensionality reduction is optimized through enhanced Principal Component Analysis (PCA), ensuring computational efficiency while retaining critical data attributes. Feature extraction encompasses a comprehensive range of metrics, including statistical measures, higher-order statistical descriptors, entropy features, and correlation-based parameters, thereby ensuring a holistic representation of customer data. At the core of this research is the advanced Evofusion model, an innovative ensemble deep learning framework. Evofusion synergizes the strengths of three neural architectures, EfficientNet, SqueezeNet, and MobileNetV2, to enhance the accuracy and robustness of customer segmentation. By leveraging the complementary capabilities of these models, Evofusion offers a sophisticated solution that surpasses the traditional methods in uncovering actionable insights. This novel approach redefines retail customer analytics by integrating cutting-edge techniques into data preprocessing, feature engineering, and ensemble modeling. This research not only advances segmentation methodologies but also provides a scalable and adaptable framework for businesses to understand and target their customer base with unprecedented precision. These results underscore the potential of deep learning ensembles to revolutionize data-driven decision-making in the retail industry..

1. INTRODUCTION

In retail, strategic customer segmentation involves grouping individuals based on shared traits, behaviors, and preferences to improve personalized interactions and marketing strategies. By analyzing factors such as purchase history, demographics, and buying habits, retailers can tailor their marketing efforts, product recommendations, and in-store experiences, thereby enhancing customer satisfaction and loyalty. The Retail Frequency Model for Tracking (RFMT) enables monitoring of shifts in customer buying patterns throughout the shopping journey [1]. The Possibilistic-Fuzzy C-Means model, which generates both memberships and possibilities alongside conventional cluster centers, improves data clustering accuracy [2]. Retailers face a surge in customer transactions, prompting the need for advanced methods to enhance segmentation precision [3]. In particular, the K-means clustering algorithm has shown superior performance in maintaining stability when seeking solutions with six clusters [4]. Understanding customer preferences is crucial when transitioning from high to low prices during product launches to ensure successful outcomes [5][6].

Efficient segmentation of unobservable utility-scale customers based solely on monthly billing data remains a challenge [7]. Many enterprises are addressing omni-channel challenges in meeting evolving customer demands, focusing on integrating online and offline operations [8]. An e-commerce segmentation model employing a three-dimensional block-matching algorithm was validated for an enhanced performance analysis [9]. Churn management in telecommunications combines churn prediction with customer segmentation for a comprehensive analytics [10]. The introduction of the "purchase tree" concept has led to "PurTreeClust," a rapid clustering algorithm designed to efficiently process customer transaction data [11]. In the industrial goods supply chain, customer segmentation helps identify varying contributions and influences mitigation strategies [12]. Predictive performance comparisons across diverse segmentation and marketing approaches have been conducted [13][14]. Artificial Intelligence (AI) is poised to revolutionize retail operations, enhancing customer satisfaction, sales, and convenience, particularly in online grocery shopping [15].

literature, Section 3 outlines the methodology, Section 4 presents the results and discussion, and Section 5 concludes the study.

2. LITERATURE REVIEW

In 2021, Tran et al. [16] integrated data science algorithms with traditional marketing strategies, incorporating the Recency, Frequency, and Monetary (RFM) model and K-Means clustering. This segmentation approach combined the Pareto/Negative binomial distribution and the Gamma-Gamma model to predict customer lifetime value (CLV) accurately. Analyzing 121,317 transactions from the bicycle retail industry, the study found that customers typically returned after three months due to the long-term usability of bicycles. The fusion of RFM segmentation with CLV assessments revealed strong correlations between well-evaluated customer segments and business value. However, a limitation was the need for regular reassessment to adapt to shifting market conditions.

In 2023, Smaili et al. [17] enhanced the segmentation model by introducing a fourth parameter, "D," which represented product diversification in a customer's purchase history. The RFM-D model employed in a retail context significantly improved the accuracy of predicting customer behavior, aiding companies in anticipating positive responses.

In the same year, Rungruang et al. [18] presented a novel hierarchical approach that integrated the RFM model with Formal Concept Analysis (FCA). This innovative clustering algorithm utilized FCA’s strengths to construct comprehensive knowledge representations, incorporating both implicit and explicit knowledge, thus improving data relationships.

Additionally, Xin et al. [19] analyzed tourism data before and during the pandemic using descriptive analytics, revealing a significant downturn in all tourism sectors, with a 70% revenue drop. The study focused on the domestic market, analyzing demographics and household expenditures to create models aimed at aiding recovery through targeted marketing strategies.

Joung et al. [20] introduced a transparent machine learning approach for customer segmentation in new product development, emphasizing product features in online reviews. The model successfully identified nonlinear relationships between feature satisfaction and overall customer contentment.

Further studies in 2023 by Bahri et al. [21], Aubery et al. [22], and Koli et al. [23] applied clustering methods like K-Means and RFM segmentation to understand consumer behavior, enhance marketing strategies, and uncover attrition patterns. T and G et al. [24] proposed using four machine learning clustering algorithms—DBSCAN, Agglomerative, K-Means, and Mean-Shift—to segment customers effectively. In 2022, Fernando et al. [25] developed a system that unified customer-product interactions and behaviors to optimize business opportunities and enhance marketing strategies.

3. RESEARCH METHODOLOGY

The research methodology for customer segmentation in the retail industry encompasses several essential steps. During the pre-processing phase, missing values are addressed using mean imputation, Z-Score Standardization is applied for normalization, One-Hot Encoding is utilized for handling categorical variables, and ADASYN (Adaptive Synthetic Sampling) is employed to resolve data imbalance. To effectively reduce dimensionality, an Improved Principal Component Analysis (PCA) is applied. Feature extraction is performed using a variety of techniques, including statistical, higher-order statistical, entropy, and correlation-based methods. The core of this approach involves customer segmentation through an ensemble deep learning strategy, integrating EfficientNet, SqueezeNet, and MobileNetV2. This robust combination of neural networks allows for the extraction of comprehensive market insights, optimizing retail customer segmentation (*see Figure 1*).

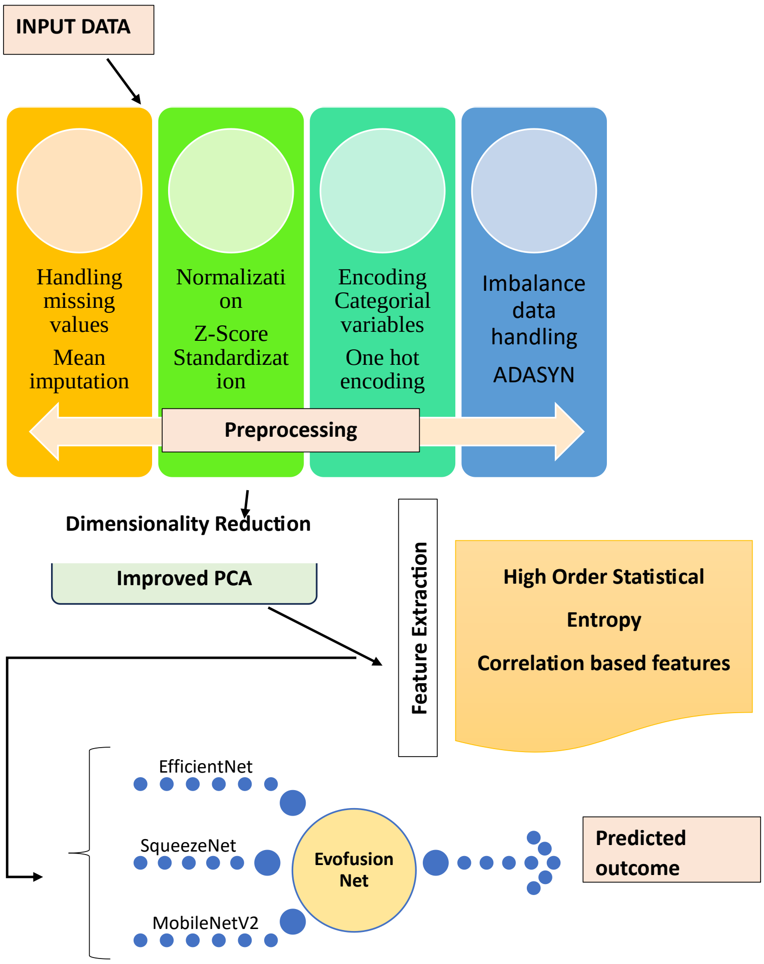

Figure 1: Work-flow diagram of the proposed model

3.1 Pre-processing

In the pre-processing phase, meticulously prepare the data for analysis. To address missing values, employ mean imputation, ensuring that our dataset is complete and suitable for analysis. For uniformity and ease of comparison, apply Z-Score Standardization to normalize the data. Categorical variables are encoded using One-Hot Encoding, allowing us to work with these variables effectively in this analysis. Additionally, handle data imbalance by implementing ADASYN (Adaptive Synthetic Sampling) to create a more balanced and representative dataset, enhancing the quality of this subsequent analyses.

3.1.1 Handling Missing Values

Mean Imputation

Addressing missing values in datasets is a fundamental aspect of data preprocessing. Mean imputation is a commonly employed technique in this regard. When dealing with missing numerical data, mean imputation involves replacing the missing values with the mean (average) value of the observed data for that variable. This method is straightforward and practical, particularly when the missingness is random and the variable's distribution is approximately normal. However, it has limitations. Mean imputation can introduce bias, especially in the presence of non-random missing data or when the data distribution is skewed. Therefore, its application should be considered within the broader context of the dataset, and alternative imputation methods may be more suitable for preserving data integrity.

3.1.2 Normalization

Normalization in customer segmentation is a data preprocessing technique vital for fair and effective segment creation. It involves transforming various customer attributes to a common scale, ensuring that no single attribute dominates the segmentation process due to its magnitude. By applying methods like Min-Max scaling or Z-score standardization, disparate data, such as income and age, are brought to a comparable range. This enables more balanced and meaningful customer segments, where each attribute contributes proportionally to the segmentation, enhancing the quality of insights and decision-making in marketing and business strategies.

Z-score Standardization

Z-score standardization, a statistical technique, plays a pivotal role in data preprocessing. This method is employed to transform data into a common scale, Standardizing data involves centering around a mean of 0 and scaling with a standard deviation of 1. This is achieved by subtracting the mean and dividing by the standard deviation., it centers the data distribution, making it convenient for comparisons and analyses, particularly when dealing with variables of different units or scales. Z-score standardization also facilitates outlier identification, offering insights into how individual data points deviate from the mean, making it a valuable tool in various fields, including finance, quality control, and research.

3.1.3 Encoding Categorized Variables

One hot encoding

One-hot encoding is a vital technique in machine learning and data analysis, transforming categorical variables into binary vectors. Particularly beneficial for nominal data lacking inherent ordinal relationships, this method creates binary columns for each unique category within a variable. For instance, if handling a "color" feature with categories like "red," "blue," and "green," one-hot encoding generates three binary columns. Each category is represented by a column, where "1" signifies the category's presence and "0" denotes its absence, facilitating effective model input.

One-hot encoding ensures that machine learning algorithms can effectively work with categorical data, as it removes the risk of them misinterpreting categorical variables as having a numerical order. However, it can increase the dimensionality of the data, which should be considered in model training. Essential for categorical variables, one-hot encoding is pivotal in boosting machine learning model accuracy by transforming categories into binary vectors.

3.1.4 Imbalance data handling

ADASYN-Adaptive Synthetic Sampling

Input:

In the context of a training dataset d_{tr} with M samples, where each sample (X_I, Y_I), $I = 1, \dots . M$ has an instance $X_I$in a Feature space X in an n-dimensional context ,and a class label Y_I in $Y = \{1, -1\}$, M_S denotes instances in the minority class, while M_L denotes instances in the majority class.. It's important to note that M_S is less than or equal to M_L, and together they sum to M.

Procedure

Step 1:The degree of class imbalance is to be calculated as

$D = M_S/M_L$ (1)

Where $D \in (0,1]$

Step 2: When D is less than D_{th} dth (a predefined threshold for class imbalance), the current class imbalance exceeds the acceptable limit.

- Determine the quantity of synthetic data required for the minority class by performing a specific calculation.

$g = (M_L - M_S) \times \beta$ (2)

The parameter β, within the range [0, 1], defines the desired balance level post-synthesis, with β = 1 achieving perfect balance.

- Compute the ratio R_I for each X_I in the minority class by identifying its K nearest neighbors in the n-dimensional space using Euclidean distance.

$R_I = \frac{\Delta_I}{k}, I = 1, \dots, M_S$ (3)

The ratio R_I falling within the range [0, 1], is determined by Δ_I, representing the count of majority class samples among X_I 's K nearest neighbors.

Normalize R_I to obtain $\widehat{R_I} = R_I / \sum_{I=1}^{M_S} R_I \ for\ i = 1\ to\ M$, ensuring $\widehat{R_I}$ forms a density distribution where $\sum_I \widehat{R_I} = 1$.

Determine the quantity of synthetic data required for each minority example X_I through specific calculations and adjustments.

$G_I = \widehat{R_I} \times g$ (4)

Here, g signifies the overall count of synthetic data instances needed for the minority class, as defined in Eq. (2)

- Produce G_I synthetic data instances for every minority class data point X_I, following the specified procedural steps.

$$Do\ the\ loop\ from\ 1\ to\ G_I$$

Select a single minority data instance, X_{ZI}, randomly from the K nearest neighbors associated with the data point X_I.

The example for synthetic data is generated by

$S_I = X_I + (X_{ZI} - X_I) \times \lambda$ (5)

λ, falling within the range [0, 1], is a random value, and $(X_{ZI} - X_I)$ represents the difference vector in an n-dimensional space.

End loop

3.2 Dimensionality Reduction

In this research, the fusion features are dimensionality reduction by the Improved PCA.

Improved Principal Component Analysis

Step 1: Initialization

Initialize the sparse loading vector β.

Choose the sparsity penalty parameter λ.

Initialize the loading vector α to zero.

Define learning rates, μ and ν, for updating α and β.

Set SLOA parameters, such as population size, maximum iterations, and other algorithm-specific parameters.

Step 2: Data processing

Import the pre-processed data matrix X with rows as data points and columns as features. Center the data by deducting feature means.

Step 3: Iterative Update

For each new data point

X_t , perform the following iterative update steps:

Step 3.1: Incremental Update of α

The goal is to update the loading vector α incrementally to fit the new data point while considering the previous estimates. The update rule can be expressed as:

$\alpha_t = \alpha_{t-1} + \mu_t \cdot X_t - (X_t \alpha_t - 1\beta_{t-1} - X_t)$ (6)

Where α_t is the updated loading vector for the sparse principal component; α_{t-1} is the previous loading vector; X_t is the new data point; μ_t is the learning rate for α at time t.

Step 3.2 Incremental update of $\boldsymbol{\beta}$.

The goal is to update the sparse component β incrementally by considering the new data point and the previous estimate. The update rue can be expressed as

$\beta_t = soft - threshold(\beta_{t-1} + v_t \cdot X_t - (X_t \alpha_{t-1} \beta_{t-1} - X_t)\lambda v_t)$ (7)

β_t is the updated sparse component; β_{t-1} is the previous sparse component; X_t is the new data point; v_t

is the learning rate for β at time t; $soft - threshold(x, \theta)$ represents the soft-thresholding function that applies the threshold θ to values of x and sets values smaller than θ to zero.

Step 4: Self-Adaptive Snow Leopard Optimization Algorithm (SLOA) for optimizing sparse component β

Apply the SLOA to optimize the sparse loading vector β. The SLOA is a population-based optimization algorithm inspired by the behavior of snow leopards in search of their prey. It includes the following steps:

Step 4.1: Initialization of Snow Leopards

Initialize a population of snow leopards, where each leopard represents a potential solution.

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_I \\ \vdots \\ x_n \end{bmatrix}_{n \times M} = \begin{bmatrix} X_{1,1} & \cdots & X_{1,D} & \cdots & X_{1,M} \\ \vdots & \ddots & \vdots & ⋰ & \vdots \\ X_{I,1} & \cdots & X_{I,D} & \cdots & X_{I,M} \\ \vdots & ⋰ & \vdots & \ddots & \vdots \\ X_{n,1} & \cdots & X_{n,D} & \cdots & X_{n,M} \end{bmatrix}_{n \times M} \quad (8)$$

In the context of snow leopard population assessment, x represents the total snow leopard population. Each x_I corresponds to an individual snow leopard, and $X_{I,D}$ signifies the value of the dth problem variable proposed by the ith snow leopard. n denotes the count of snow leopards in the algorithm population, while M indicates the total number of problem variables under consideration.

Step 4.2: Objective Function Evaluation

Evaluate the fitness of each snow leopard in the population based on the quality of the sparse principal component represented by its β.

Step 4.3 Constraint Handling

Implement a constraint-handling mechanism to ensure that solutions in the population satisfy any problem-specific constraints.

Identify infeasible solutions and repair or penalize them appropriately to make them feasible.

Use penalty functions, repair methods to handle constraints and guide infeasible solutions toward feasibility.

Nonlinear constrained optimization aims to discover feasible solutions that meet all specified constraints. The conventional formulation of such a problem typically involves finding optimal values for decision variables within the constraints' boundaries to minimize or maximize an objective function.

$$minimize \quad f(X) subject\ to$$

$G_J(X) \leq 0, J = 1$ (9)

$H_K(X) = 0, \ \ K = 1$ (10)

$X_I^l \leq X_I \leq X_I^U, \ I = 1$ (11)

In the context of optimization with n variables, J inequality constraints G_J, and K equality constraints $H_K(X)$, the conventional approach involves transforming equality constraints into equivalent inequality constraints, where two inequality constraints are substituted for each equality constraint $H_K(X)$.: $H_K(X) \geq 0$ and $-H_K(X) \geq 0$. This simplifies the overall constraint handling process. Each variable's range: $[X_I^l, \ X_I^U]$.

Step 4.4 ---Phase 1: Routes of Travel and Movement

Snow Leopards move in a zig-zag pattern and use scent signs.

This phase simulates how snow leopards update their locations to efficiently search the space.

Snow leopards employ scent signs, a common feline behavior, to mark their territories and pathways. Before marking territory, snow leopards often scrape the ground with hind feet. Their movement features a zig-zag pattern, opting for indirect routes, influencing their interactions . The proposed Snow Leopard Observation Algorithm (SLOA) mathematically captures this behavior in Equations (12)-(14).

$$X_{I,D}^{P1} = X_{I,D} + R \times \left(X_{K,D} - I \times X_{I,D}\right) \times sign(f_I - f_K), K\epsilon 1,2,3, \dots n; D = 1,2,3, \dots M \qquad (12)$$

$$x_I = \begin{cases} x_I^{P1}, & f_I^{P1} < f_I \\ x_{I,} & else \end{cases} \qquad (13)$$

$$I = round(1 + R) \qquad (14)$$

In this scenario, $X_{I,D}^{P1}$signifies the updated value for the Dth problem variable determined by the Ith snow leopard during phase 1. The variable R represents a random number in the [0, 1] interval, and K denotes the row number of the guiding snow leopard for the ith individual along the Dth axis. Meanwhile, x_I^{P1} reflects the revised position of the ith snow leopard in phase 1, and f_I^{P1} indicates its objective function value.

Phase 2: Hunting process

Snow leopards use rocky cliffs to approach prey.

The algorithm simulates how snow leopards move toward prey and update their positions after an attack.

Phase 3: Reproduction process

This stage simulates the mating habits and reproductive behavior characteristic of snow leopards in their natural environment. New members are added to the population based on the mating of two snow leopards.

Continuously observe the population and dynamically adjust the mutation rate ($p_{mutation}$) and crossover rate ($p_{crossover}$).

Adaptive Mutation operator: The Adaptive Mutation Operator allows for dynamic adjustment of the mutation rate, employing a strategy such as Gaussian mutation. The mutation strength, denoted by σ, adapts in response to the algorithm's progress. The formula for adaptive mutation can be expressed as follows:

$$\sigma(t) = \sigma_o \cdot \exp(-T_t) \qquad (15)$$

Where $\sigma(t)$ is the mutation strength at iteration t; σ_o is the initial mutation strength; T is a time constant that affects how quickly σdecreases. Can experiment with different values for T.

Adaptive Crossover Operator:

To implement adaptive crossover, you can modify the crossover rate, represented as $p_{crossover}(t)$, based on the algorithm's performance at iteration t. The adjustment of crossover $p_{crossover}$ is determined by continuously assessing population diversity and convergence speed. The formula for adaptive crossover is expressed as follows:

$$p_{crossover}(t) = p_{crossover} \cdot (1 - D_{max} D(t)) (16)$$

Where $p_{crossover}(t)$ is the crossover rate at iteration t. $p_{crossover}$is the initial crossover rate.

$D(t)$ is a measure of population diversity at iteration t.

- $maxD_{max}$ is the maximum allowed population diversity. If $D(t)$ is close to $maxD_{max}$, the crossover rate decreases

The choice of specific measures for diversity ($D(t)$) and time constants may depend on the problem and the adaptation strategy you wish to implement.

Phase 4: Mortality

Snow leopards may face mortality, and the algorithm simulates the loss of members.

Snow leopards exhibiting weaker objective function values are at a higher likelihood of facing adverse consequences or mortality.

Cubs born to snow leopards may face mortality if they exhibit suboptimal objective function values, impacting their survival prospects.

Step 4.5: Convergence Criteria

Check for convergence based on predefined criteria, such as a maximum number of iterations or the improvement in fitness values.

Step 5: Repeat for All Data Points

- Iterate Steps 3.1 and 3.2 for each new data point X_t.

3.3 Feature Extraction

Statistical

Statistical features are indispensable for customer segmentation, as they enable businesses to dissect their customer base effectively. These features encompass a spectrum of metrics that offer insights into customer behavior. Among these, metrics like purchase frequency help identify how often customers buy, while average transaction value highlights the mean amount spent during each purchase. Recency measures how recently a customer made a purchase, providing clues about their engagement. Customer lifetime value measures the cumulative worth a customer contributes throughout their entire association with the company, gauging long-term value. Basket size examines the number of items in a customer's cart, and churn rate tracks the percentage of customers who cease making purchases. By leveraging these statistical features, businesses can form distinct customer segments and tailor marketing strategies to meet the unique needs of each group, ultimately enhancing customer satisfaction and loyalty.

Higher order statistical

In data analysis and feature engineering, advanced statistical measures like higher-order statistical features, entropy, and correlation-based features play a pivotal role. Higher-order statistics delve deeper into data distributions, while entropy quantifies data uncertainty. Correlation-based features explore relationships between variables, offering insights into dependencies. These metrics are invaluable for uncovering hidden patterns, aiding in diverse fields such as image recognition, financial analysis, and machine learning.

Entropy

Entropy, a fundamental concept in information theory and statistics, quantifies the uncertainty or randomness of a dataset. It measures the amount of disorder or unpredictability within data, with higher entropy indicating greater randomness. In applications like data compression, machine learning, and cryptography, understanding and manipulating entropy are essential for effective information processing and security.

Correlation based features

In the context of customer segmentation, correlation-based features are instrumental for identifying patterns and dependencies among customer attributes. By measuring the strength and direction of relationships between variables such as purchase history, demographic data, and customer preferences, these features help segment customers effectively. For instance, they can uncover associations between age and purchase frequency or income and product preferences, allowing businesses to create more refined and targeted customer segments. This enhances marketing strategies, product recommendations, and customer experiences, ultimately leading to improved customer satisfaction and loyalty.

3.4 Customer Segmentation via Evo fusion model

Efficient Net

Efficient Net represents a groundbreaking convolutional neural network architecture founded on the concept of "compound scaling." This innovative approach effectively tackles the age-old challenge of balancing model size, computational efficiency and accuracy. The principle driving compound scaling centers on the dynamic adjustment of three key dimensions within the neural network: width, depth, and resolution.

Width-Scaling: Width scaling focuses on amplifying the channel count in network layers. Augmenting width bolsters a network's capability to grasp intricate patterns, elevating accuracy. In contrast, narrowing width streamlines the model, adapting it for resource-constrained environments, optimizing efficiency without compromising performance in scenarios with limited computational resources.

Depth-Scaling: Depth_scaling focuses on tailoring the network's layer count to specific requirements. Deeper networks excel at capturing intricate data representations, albeit at the cost of higher computational resources. Conversely, shallower models prioritize computational efficiency, although they may sacrifice some degree of Accuracy.

Resolution Scaling: Resolution scaling involves modifying the input image size. Elevated resolutions offer more detailed information, potentially enhancing performance. However, this comes at the cost of heightened memory and computational requirements. It is a trade-off between improved visual fidelity and the practical constraints of memory and computational resources (*see Figure 2*).

Lower-resolution inputs, conversely, are more resource-efficient but may compromise fine-grained detail [26].

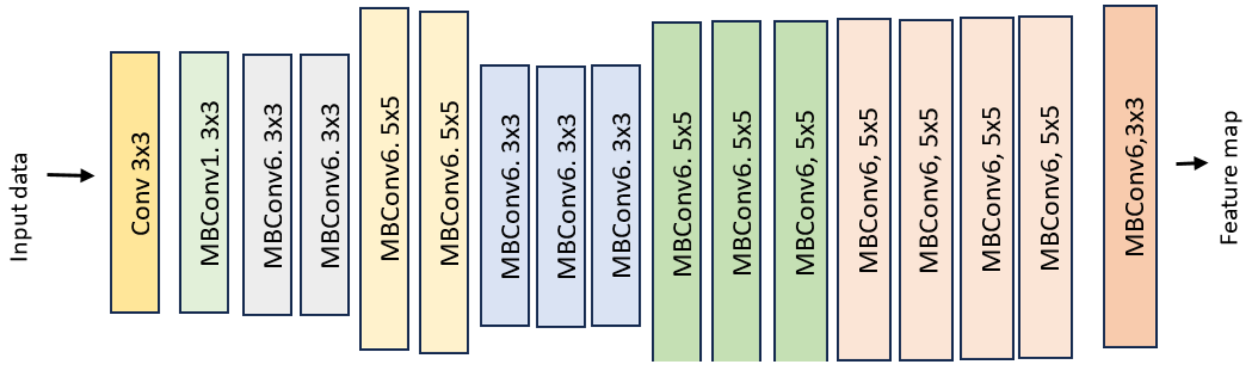

Figure 2: Efficient Net architecture

Squeeze Net

In an effort to minimize parameter count, researchers in [36] devised Squeeze Net, a convolutional neural network. Unlike traditional 3x3 filters, Squeeze Net primarily employs 1x1 filters in its layers, significantly reducing the number of parameters—by a factor of 9. Additionally, it introduces down-sampling of large activation maps to enhance classification accuracy. Squeeze Net is structured as a series of consecutive fire modulesWithin this architecture, every fire module comprises a squeeze convolution layer, solely utilizing 1x1 filters, succeeded by an expand layer that integrates a combination of 1x1 and 3x3 filters [27]. This design enhances the network's capacity for feature extraction and contributes to its overall efficiency. Similar to Res Net, Squeeze Net maintains consistent connectivity between layers and fire modules. Despite its efficiency, Squeeze Net only comprises 1.24 million parameters, making it a lightweight backbone compared to other networks (*see Figure 3*).

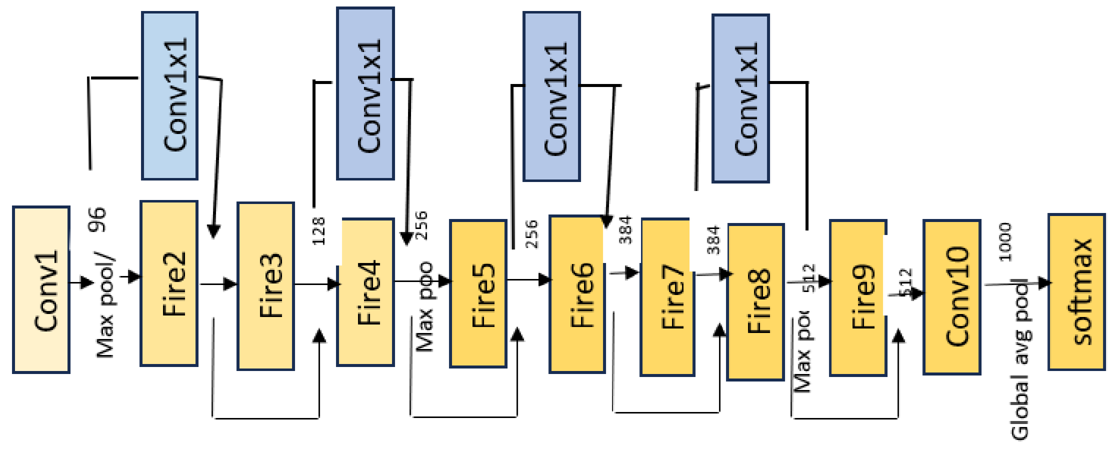

Figure 3: Squeeze Net architecture

MobileNetV2

While MobileNetV2 is primarily known for its applications in computer vision and image-related tasks, it can also be adapted for customer segmentation within the context of mobile and edge computing. Its efficiency and flexibility make it a viable option for analyzing customer data and creating segments based on various attributes. With its inverted residual blocks, depth wise separable convolutions, and the ability to adjust model size using width and resolution multipliers, MobileNetV2 can efficiently process customer data while conserving computational resources (*see Figure 4*). This adaptability allows businesses to perform customer segmentation tasks effectively, enabling targeted marketing strategies, personalized recommendations, and improved customer experiences, all on mobile and edge devices [28].

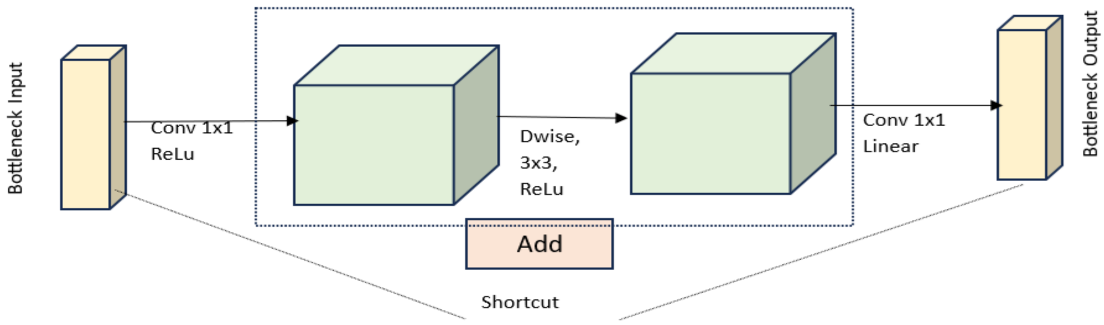

Figure 4: MobileNetV2 architecture

Convolutional Layer

A convolutional layer in deep learning is a fundamental component of convolutional neural networks. Executing convolution on input data involves sliding a small filter (kernel) to extract features, capturing spatial relationships and enhancing information extraction. This layer is crucial for image and spatial data analysis, enabling the network to automatically learn and identify patterns, edges, and textures within the data.

Max Pooling layer

A max pooling layer is an essential element in a convolutional neural network used for down-sampling and feature selection. It operates on a grid of values produced by a convolutional layer and reduces the spatial dimensions by selecting the highest value in each region. This process retains the most significant features while reducing computational complexity and preventing overfitting. Max pooling helps extract robust features for image and data analysis tasks.

Fully Connected layer

A dense layer, synonymous with a fully connected layer in neural networks. In this layer, every neuron is connected to every neuron in the previous and subsequent layers. It plays a role in capturing complex, high-level relationships in the data and is often used in the final layers of deep learning models for tasks like classification. The layer's weights are learned during training, enabling it to model intricate patterns in the data**.**

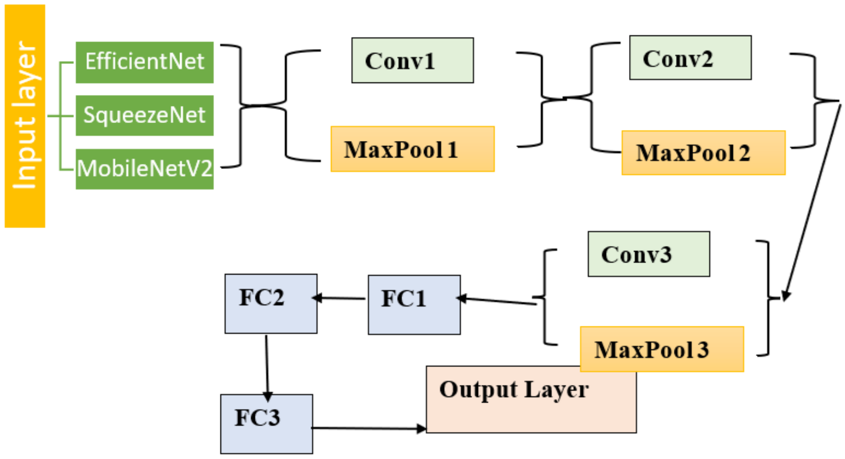

Figure 5: Evofusion Model

The Evo Fusion model employs a sophisticated architecture for customer segmentation, seamlessly combining Efficient Net, Squeeze Net, and MobileNetV2. The multi-stage design includes Convolutional Layers, Fully Connected Layers and Max Pooling Layers, forming a deep neural network. This intricate structure enables the model to extract hierarchical features from input data, leading to refined customer segmentation results (*see Figure 5*).

4. RESULT AND DISCUSSION

In this research, the Evo fusion model is implemented in PYTHON, incorporating a range of performance metrics such as Precision, Accuracy, F-Measure, Sensitivity, Specificity, NPV, FPR, MCC, and FNR. The objective is to optimize accuracy, exploring learning rates at 70% and 80% for comprehensive performance evaluation and model refinement.

In *Table 1*, the performance metrics at a 70% learning rate reveal distinctive outcomes for various models. Notably, the Evo fusion model showcases superior results across multiple measures due to the usage of improved PCA and self-adaptive snow leopard optimization, attaining an accuracy of 97.86%, precision at 98.25%, sensitivity of 97.57%, specificity reaching 98.17%, and F-measure achieving 97.91%. Additionally, the Matthews Correlation Coefficient (MCC) demonstrates a notable value of 95.72%, emphasizing the robustness of the proposed model. Other metrics, including Negative Predictive Value , False Positive Rate, and False Negative Rate, further underscore the Evo fusion model's efficacy in comparison to benchmarks like Efficient Net, Squeeze Net, Mobile Net, and Res Net.

Table 1: Performance metrics of learning rate 70%

Metrics	Efficient Net	Squeeze Net	Mobile Net	Res Net	PROPOSED (Evo fusion)
Accuracy	0.9664607	0.9129887	0.9649805	0.939567	0.97861
Precision	0.9626998	0.9197995	0.9702381	0.941964	0.982517
Sensitivity	0.9695886	0.9084158	0.9588235	0.939866	0.975694
Specificity	0.9634146	0.9177378	0.9710425	0.939252	0.981685
F-Measure	0.9661319	0.9140722	0.964497	0.940914	0.979094
MCC	0.9329392	0.8260222	0.9300139	0.879073	0.957221
NPV	0.9701754	0.9060914	0.9599237	0.937063	0.974545
FPR	0.0365854	0.0822622	0.0289575	0.060748	0.018315
FNR	0.0304114	0.0915842	0.0411765	0.060134	0.024306

In *Table 2*, assessing the performance metrics at an 80% learning rate unveils distinct outcomes for various models. Notably, the Evo fusion model excels, achieving a remarkable accuracy of 99.03%, precision at 99.31%, sensitivity of 98.79%, specificity reaching 99.28%, and F-measure attaining 99.05%. The Matthews Correlation Coefficient (MCC) demonstrates exceptional value at 98.06%, underscoring the robustness of the proposed model. Other metrics, including Negative Predictive Value, False Positive Rate, and False Negative Rate, further emphasize the Evo fusion model's superior efficacy in comparison to benchmarks like Efficient Net, Squeeze Net, Mobile Net, and Res Net at an 80% learning rate. These all due to the usage of Improved PCA and Self Adaptive snow leopard optimization algorithm.

Table 2: Performance metrics-learning rate 80%

Metrics	EfficientNet	SqueezeNet	MobileNet	ResNet	PROPOSED
Accuracy	0.97320574	0.93194925	0.96545455	0.9490791	0.99028269
Precision	0.97495183	0.93424036	0.96022727	0.9491525	0.99305556
Sensitivity	0.97120921	0.9321267	0.96755725	0.9511677	0.98791019
Specificity	0.97519084	0.93176471	0.96354167	0.9469027	0.99276673
F_Measure	0.97307692	0.93318233	0.96387833	0.9501591	0.99047619
MCC	0.94641739	0.86385461	0.93080291	0.8981126	0.98057127
NPV	0.97148289	0.92957746	0.97027972	0.9490022	0.98741007
FPR	0.02480916	0.06823529	0.03645833	0.0530973	0.00723327
FNR	0.02879079	0.0678733	0.03244275	0.0488323	0.01208981

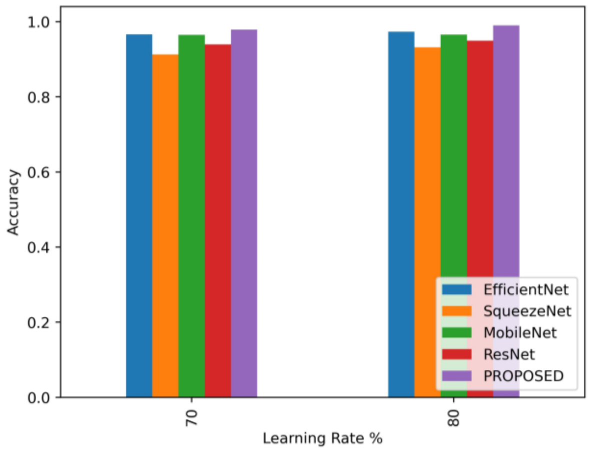

Figure 6: Performance metrics of the proposed model accuracy with

learning rate 70% and 80%

Figure 6 illustrates the performance metrics Accuracy of the proposed model with learning rate 70% and 80%. It is clear that the suggested model Evo fusion accuracy is high when compared to the other models due to the improved PCA and the self Adaptive snow leopard optimization algorithm

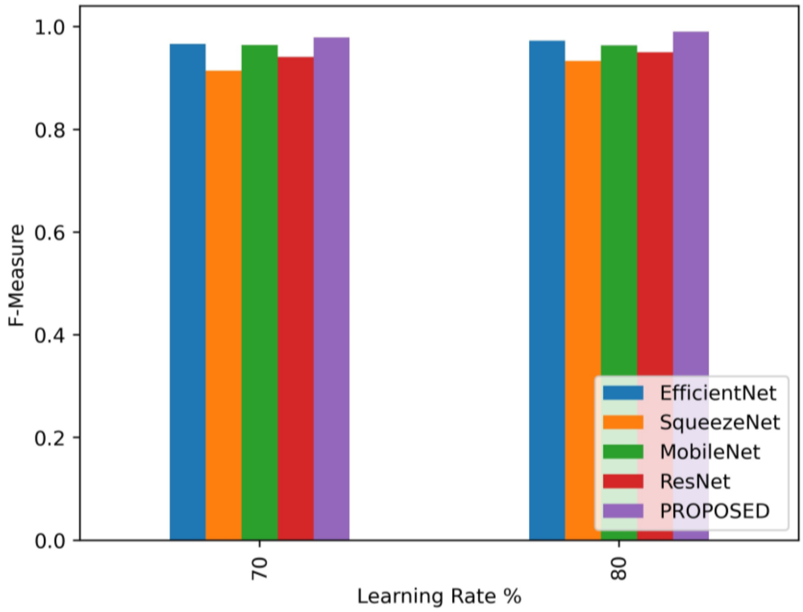

Figure 7: Performance metrics of the proposed model F-measure with

learning rate 70% and 80%

Figure 7 illustrates the performance metrics F-measure of the proposed model with learning rate 70% and 80%. It is clear that the proposed model Evo fusion F-Measure is high when compared to the other models due to the improved PCA and the Self Adaptive snow leopard optimization algorithm.

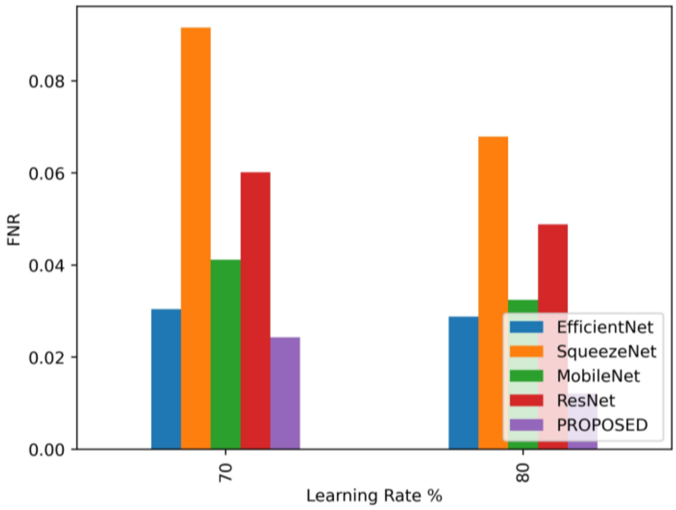

Figure 8: Performance metrics of the proposed model FNR with

learning rate 70% and 80%

Figure 8 demonstrates the performance metrics FNR of the suggested model with learning rate 70% and 80%. It is clear that the proposed model Evo fusion FNR is low when compared to the other models due to the improved PCA and the Self Adaptive snow leopard optimization algorithm.

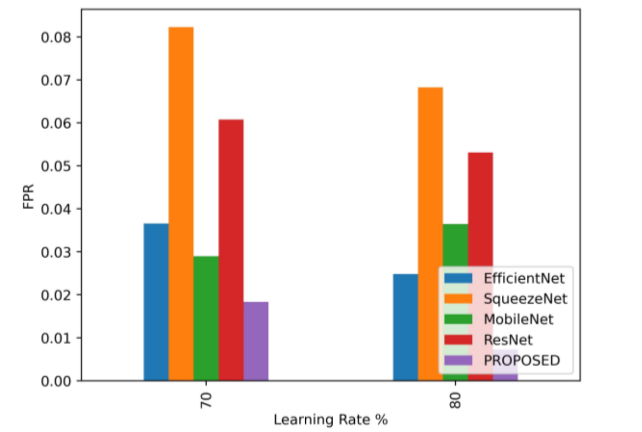

Figure 9: Performance metrics of the proposed model FPR with

learning rate 70% and 80%

Figure 9 illustrates the performance metrics FPR of the proposed model with learning rate 70% and 80%. It is clear that the proposed model Evo fusion FPR is low when compared to the other models due to the improved PCA and the self Adaptive snow leopard optimization algorithm.

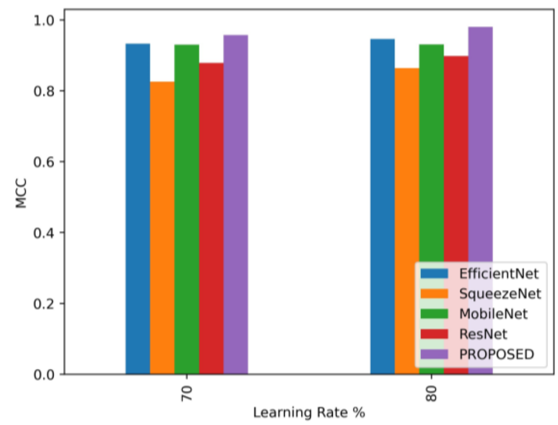

Figure 10: Performance metrics of the proposed model MCC with

learning rate 70% and 80%

Figure 10 illustrates the performance metrics MCC of the proposed model with learning rate 70% and 80%. It is clear that the proposed model Evo fusion MCC is high when compared to the other models due to the improved PCA and the Self Adaptive snow leopard optimization algorithm.

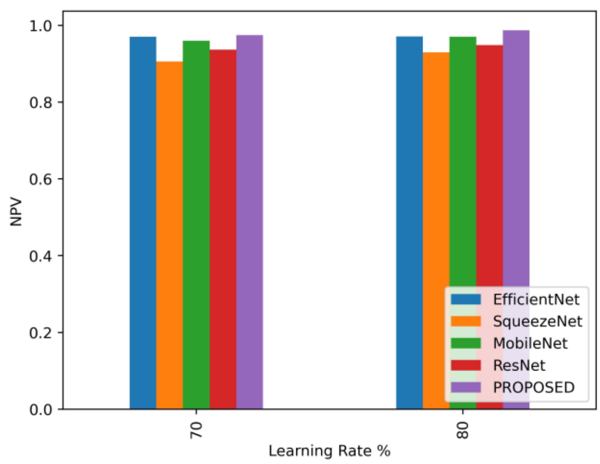

Figure 11: Performance metrics of the proposed model NPV with

learning rate 70% and 80%

Figure 11 illustrates the performance metrics NPV of the proposed model with learning rate 70% and 80%. It is clear that the proposed model Evo fusion NPV is little high when compared to the other models due to the improved PCA and the self Adaptive snow leopard optimization algorithm.

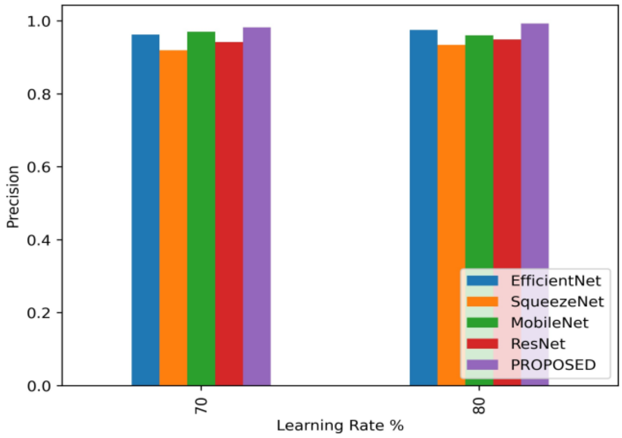

Figure 12: Performance metrics of the proposed model precision with

learning rate 70% and 80%

Figure 12 illustrates the performance metrics Precision of the proposed model with learning rate 70% and 80%. It is clear that the proposed model Evo fusion Precision is high when compared to the other models due to the improved PCA and the self Adaptive snow leopard optimization algorithm.

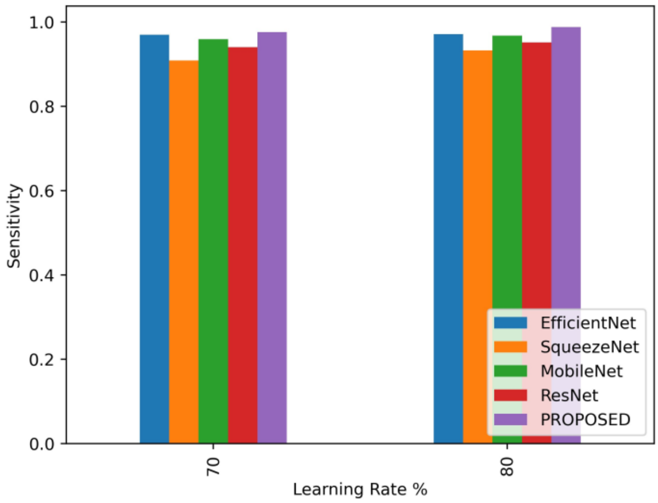

Figure 13: Performance metrics of the proposed model sensitivity with

learning rate 70% and 80%

Figure 13 illustrates the performance metrics Sensitivity of the proposed model with learning rate 70% and 80%. It is clear that the proposed model Evo fusion Sensitivity is high when compared to the other models due to the improved PCA and the Self Adaptive snow leopard optimization algorithm.

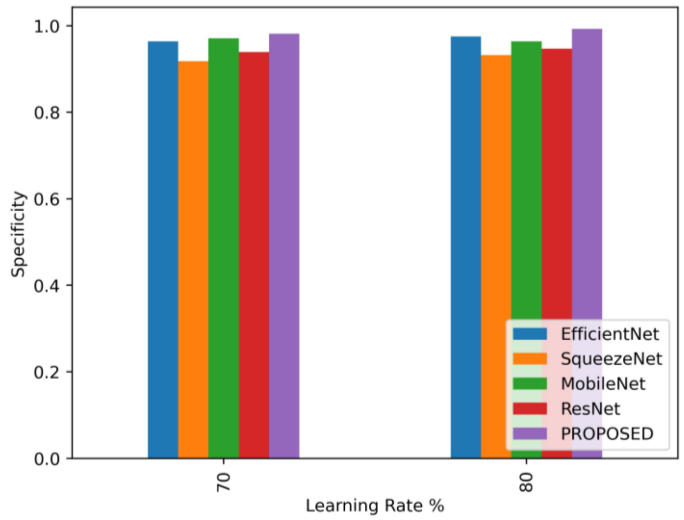

Figure 14: Performance metrics of the proposed model Specificity with

learning rate 70% and 80%

Figure 14 illustrates the performance metrics Specificity of the proposed model with learning rate 70% and 80%. It is clear that the proposed model Evo fusion Specificity is high when compared to the other models due to the improved PCA and the Self Adaptive snow leopard optimization algorithm.

5. CONCLUSION

In conclusion, this study focused on retail analytics by exploring and comparing various clustering algorithms for effective customer segmentation. The data preprocessing procedures, including mean imputation, Z-score standardization, and one-hot encoding, were critical in ensuring high data quality. To address data imbalances, ADASYN was employed, while dimensionality reduction was enhanced using an improved version of Principal Component Analysis, further optimizing model performance. The feature extraction process incorporated a broad range of statistical techniques to extract meaningful patterns. The Evo fusion model, which leverages an ensemble deep learning approach combining EfficientNet, SqueezeNet, and MobileNetV2 in Python, was implemented as a powerful solution for customer segmentation. This comprehensive framework offers valuable insights for advancing retail analytics and understanding the ever-evolving dynamics of the retail market..

REFERENCES

[1] Zhou, J., Wei, J. and Xu, B., 2021. Customer segmentation by web content mining. Journal of Retailing and Consumer Services, 61, p.102588.

[2] Kuo, R.J., Alfareza, M.N. and Nguyen, T.P.Q., 2023. Genetic based density peak possibilistic fuzzy c-means algorithms to cluster analysis-a case study on customer segmentation. Engineering Science and Technology, an International Journal, 47, p.101525.

[3] John, J.M., Shobayo, O. and Ogunleye, B., 2023. An Exploration of Clustering Algorithms for Customer Segmentation in the UK Retail Market. Analytics, 2(4), pp.809-823.

[4] Wilbert, H.J., Hoppe, A.F., Sartori, A., Stefenon, S.F. and Silva, L.A., 2023. Recency, Frequency, Monetary Value, Clustering, and Internal and External Indices for Customer Segmentation from Retail Data. Algorithms, 16(9), p.396.

[5] Tavor, T., Gonen, L.D. and Spiegel, U., 2023. Customer Segmentation as a Revenue Generator for Profit Purposes. Mathematics, 11(21), p.4425.

[6] Wilbert, H.J., Hoppe, A.F., Sartori, A., Stefenon, S.F., Silva, L.A. and Leithardt, V.R.Q., 2023. Using Clustering for Customer Segmentation from Retail Data.

[7] Y. Yuan, K. Dehghanpour, F. Bu and Z. Wang, "A Data-Driven Customer Segmentation Strategy Based on Contribution to System Peak Demand," in IEEE Transactions on Power Systems, vol. 35, no. 5, pp. 4026-4035, Sept. 2020, doi: 10.1109/TPWRS.2020.2979943.

[8] H. -H. Zhao, X. -C. Luo, R. Ma and X. Lu, "An Extended Regularized K-Means Clustering Approach for High-Dimensional Customer Segmentation With Correlated Variables," in IEEE Access, vol. 9, pp. 48405-48412, 2021, doi: 10.1109/ACCESS.2021.3067499.

[9] F. Liu, "3D Block Matching Algorithm in Concealed Image Recognition and E-Commerce Customer Segmentation," in IEEE Sensors Journal, vol. 20, no. 20, pp. 11761-11769, 15 Oct.15, 2020, doi: 10.1109/JSEN.2019.2936169.

[10] S. Wu, W. -C. Yau, T. -S. Ong and S. -C. Chong, "Integrated Churn Prediction and Customer Segmentation Framework for Telco Business," in IEEE Access, vol. 9, pp. 62118-62136, 2021, doi: 10.1109/ACCESS.2021.3073776.

[11] X. Chen, Y. Fang, M. Yang, F. Nie, Z. Zhao and J. Z. Huang, "PurTreeClust: A Clustering Algorithm for Customer Segmentation from Massive Customer Transaction Data," in IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 3, pp. 559-572, 1 March 2018, doi: 10.1109/TKDE.2017.2763620.

[12] Q. Gao, D. Xia, Y. Shi and J. Quan, "Policies Adoption for Supply Disruption Mitigation Based on Customer Segmentation," in IEEE Access, vol. 7, pp. 47329-47338, 2019, doi: 10.1109/ACCESS.2019.2905547.

[13] Tianyi Jiang and A. Tuzhilin, "Segmenting Customers from Population to Individuals: Does 1-to-1 Keep Your Customers Forever?," in IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 10, pp. 1297-1311, Oct. 2006, doi: 10.1109/TKDE.2006.164.

[14] M. Espinoza, C. Joye, R. Belmans and B. De Moor, "Short-term load forecasting, profile identification, and customer segmentation: a methodology based on periodic time series," in IEEE Transactions on Power Systems, vol. 20, no. 3, pp. 1622-1630, Aug. 2005, doi: 10.1109/TPWRS.2005.852123.

[15] Aggarwal, D., Exploring the Role of AI for Online Grocery Shopping through Enhancing Personalized Recommendations and Customer Segmentation.

[16] Tran, K.G., Nguyen, V.H. and Ho, T., Customer segmentation analysis and customer lifetime value prediction using Pareto/NBD and Gamma-Gamma model in 2021.

[17] Smaili, M.Y. and Hachimi, H., 2023. New RFM-D classification model for improving customer analysis and response prediction. Ain Shams Engineering Journal, p.102254.

[18] Rungruang, C., Riyapan, P., Intarasit, A., Chuarkham, K. and Muangprathub, J., 2023. RFM model customer segmentation based on hierarchical approach using FCA. Expert Systems with Applications, 237, p.121449.

[19] Xin, L.Y., Yiteng, L., Jain, R., Van, T.T.H., Lim, W. and Yilin, Z., 2023. Marketing Segmentation and Targeted Marketing for Tourism. In Tourism Analytics Before and After COVID-19: Case Studies from Asia and Europe (pp. 139-155). Singapore: Springer Nature Singapore.

[20] Joung, J. and Kim, H., 2023. Interpretable machine learning-based approach for customer segmentation for new product development from online product reviews. International Journal of Information Management, 70, p.102641.

[21] R. S. Bahri, I. D. Sudirman, I. D. Utama and R. H. Susanto, "Data Mining Techniques To Uncovering Customer Segments: K-Means Clustering Using The Elbow Method Approach In Medium-Scale Grocery," 2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE), Jakarta, Indonesia, 2023, pp. 262-266, doi: 10.1109/ICCoSITE57641.2023.10127826.

[22] P. Aubrey, A. A. Permana and J. C. Young, "Implementation of Recency Techniques Monetary and K-Means for the Consumer Segmentation System," 2023 International Conference on Smart Computing and Application (ICSCA), Hail, Saudi Arabia, 2023, pp. 1-6, doi: 10.1109/ICSCA57840.2023.10087563.

[23] S. Koli, R. Singh, R. Mishra and P. Badhani, "Imperative role of customer segmentation technique for customer retention using machine learning techniques," 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), Greater Noida, India, 2023, pp. 243-248, doi: 10.1109/AISC56616.2023.10085487.

[24] M. T, S. G and M. A, "A Machine Learning Approach to Segment the Customers of Online Sales Data for Better and Efficient Marketing Purposes," 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-9, doi: 10.1109/ICECONF57129.2023.10084339.

[25] A. M. P. Fernando, A. M. T. T. Adhikari, W. H. A. T. K. Wijesekara, T. V. T. I. Vithanage, A. Gamage and T. Jayalath, "Planning Marketing Strategies in Small-Scale Business Using Data Analysis," 2022 3rd International Informatics and Software Engineering Conference (IISEC), Ankara, Turkey, 2022, pp. 1-6, doi: 10.1109/IISEC56263.2022.9998200.

[26] https://images.app.goo.gl/Fosq26iyeereuSBS9

[27] https://images.app.goo.gl/rnVW2mUDs6HsTCMB6

[28] https://images.app.goo.gl/QJcNWE8XCQoPDVWB8

❖❖❖❖❖