Optimizing Segmentation and Purchase Forecasting in Credit Card Transactions: A PSO-enhanced K-means and ANN Approach

Shintami Chusnul Hidayati[1], Putu Bagus Gede Prasetyo Raharja[2], I Nyoman Gde Artadana Mahaputra Wardhiana[3], Sebastian Klemm[4]

[1,2,3,4] Department of Informatics, Institut Teknologi Sepuluh Nopember
[1]shintami@its.ac.id
[2] 6025231010@student.its.ac.id
[3]6025231019@student.its.ac.id
[4]5999231036@student.its.ac.id

***Abstract*—In the rapidly evolving landscape of data-driven marketing, machine learning has emerged as a pivotal tool for analyzing complex consumer behaviors and enhancing strategic decision-making. This paper introduces a novel approach to optimize customer segmentation and purchase forecasting in credit card transactions through the synergistic integration of Particle Swarm Optimization (PSO)-enhanced k-means clustering and Artificial Neural Networks (ANN). The proposed methodology refines customer segmentation by leveraging PSO, resulting in more defined clusters. In the predictive modeling phase, an ANN outperforms conventional methods, providing superior accuracy in purchase forecasting. The study demonstrates the effectiveness of advanced algorithms in enhancing insights from credit card transaction data, offering valuable implications for improved decision-making in the financial domain.**

***Kata Kunci*— credit card transaction, customer segmentation, neural network, optimized k-means, particle swarm optimization, purchase forecasting.**

I. Introduction

In the dynamic landscape of finance and data analytics, credit card transactions emerge as a robust source of information, unveiling valuable insights into consumer behavior and financial trends. The strategic acquisition of new customers and the retention of profitable ones stand as critical objectives for financial institutions and banks [1]. As the volume and intricacy of credit card data continue to surge, the imperative to leverage advanced techniques becomes paramount, facilitating the extraction of meaningful patterns and the enhancement of decision-making processes. Recognizing the inherent diversity in customers' financial performances, there arises a need for distinct treatment strategies based on repayment and purchasing behavior. Customers exhibiting such behaviors can be incentivized and rewarded for their loyalty and profitability [2]. However, the challenge lies in the non-automated procedure involved in customer segmentation and behavioral prediction, spanning from credit score calculation to delinquent prediction and preventive measures.

Previous studies have showcased the effectiveness of various machine learning models, including K-nearest Neighbor, Decision Tree, Random Forest Classifier, Logistic Regression, Support Vector Machine, XGBoost, and Naive Bayes, in predicting customer behavior based on credit card data [3] [4]. Machine learning-based approaches, in contrast to manual operations, not only require less time to complete but also produce more accurate results. The field of machine learning is fundamentally concerned with algorithms that empower computers to learn and grow through experience without explicit programming [5]. This involves constructing a model using training data, evaluating it, and then combining algorithms and statistical models to enable computers to perform tasks without the need for hard coding [6] [7].

Motivated by the above reasons, this paper delves into the realm of applying machine learning methodologies to optimize two pivotal aspects of credit card data analysis: customer segmentation and purchase forecasting. The approach seeks to proficiently group clients based on repayment habits and capture relationships to predict purchase behavior. Utilizing a public dataset containing pertinent information on credit card users, we explore an innovative approach that integrates Particle Swarm Optimization (PSO)-enhanced k-means clustering and Artificial Neural Networks (ANN). Findings based on experiments on a public dataset underscore the importance of leveraging advanced algorithms for data processing, clustering, and predictive modeling in enhancing the precision of customer-related tasks. By synergizing the power of optimization algorithms with the adaptability of neural networks, this study aims to contribute to the ongoing discourse on the strategic utilization of machine learning in financial domains.

The rest of this section is organized as follows. Section II covers related work. Section III details the dataset. Section IV provides detailed of the proposed approach. Section V discusses experimental results. Finally, Section VI concludes the paper and suggests possible future research directions.

II. Related Work

In the domain of banking, particularly credit card data analysis, machine learning models have proven indispensable for enhancing decision-making processes. Wu *et al.* [8] conducted credit risk assessment in the Chinese credit card industry, introducing the innovative Deep Multiple Kernel Classifier (DMKC). By integrating deep learning with multiple kernel learning, the DMKC demonstrated superior performance over conventional models in handling complex and high-dimensional data, offering a more efficient tool for credit risk assessment.

Addressing the pressing concern of credit card fraud, Xueping *et al*. [4] focused on the growing challenges posed by mobile web applications and electronic payment systems. Their study employed a combination of XGBoost and Multi-Layer Perceptron (MLP) models, revealing the effectiveness of XGBoost in fraud detection. Through extensive experiments on a public dataset, they provided empirical evidence supporting the superior performance of XGBoost over MLP in addressing credit card fraud issues.

In a bid to understand and predict customer behavior based on credit card data, Arora *et al*. [3] concentrated on the challenge of credit card defaults in the banking sector. Utilizing a diverse set of machine learning algorithms, including K-nearest Neighbor, Decision Tree, Random Forest Classifier, Logistic Regression, Support Vector Machine, and Naive Bayes, the study aimed to identify the most effective technique for predicting defaults. Their comprehensive approach, involving data preprocessing, analysis, and algorithm application, contributes valuable insights for financial institutions engaged in risk management. These studies collectively underscore the pivotal role of machine learning in addressing various challenges within the realm of credit card transactions and banking.

In contrast to the tasks mentioned earlier, we investigated the application of machine learning methodologies to enhance two vital aspects of credit card data analysis: customer segmentation and purchase forecasting. To achieve this, we introduced optimizations to a widely adopted approach, exploring their advantages in terms of convergence, adaptability, and generalization.

III. Dataset

This study utilizes the publicly available Kaggle dataset "Credit Card Data for Clustering" [1], which captures the behavioral patterns of approximately 9,000 active credit card holders over the past 6 months in 2017. With 17 behavioral variables, including customer identification (*cust_id*), the dataset provides a comprehensive overview of credit card usage. All features, except *cust_id*, are expressed in numerical values, each with a distinct scale (see Table I for details).

In our experiments, we partition the data into two sets: training and test sets. The training set, comprising 80% of the data, is used to train and fine-tune our models, allowing them to learn and adapt to the patterns and intricacies within the data. The remaining 20%, the test set, is used in evaluating how well the model generalizes to new, unseen data.

IV. Proposed Approach

In this section, we begin by outlining the data processing approach employed to enhance data quality. Subsequently, we provide a detailed explanation of the methods used for customer segmentation through clustering and the approach utilized for forecasting purchases.

A. Data Preprocessing

In this phase, we enhance data quality by addressing missing values, eliminating duplicates, and normalizing features to establish a consistent scale. This preprocessing optimizes the dataset for subsequent machine learning algorithms, forming the basis for precise customer segmentation and purchase prediction.

TABLE I
DATASET FEATURES AND ITS DESCRIPTION

Features	Description
cust_id	Identification of Credit Card holder (Categorical)
balance	Balance amount left in their account to make purchases
balance_frequency	Balance update frequency is represented by a score between 0 and 1
purchases	Amount of purchases made from account
oneoff_purchases	Maximum purchase amount done in one-go
installments_purchases	Amount of purchase done in installment
cash_advance	Cash in advance given by the user
purchases_frequency	Purchase frequency, scored between 0 and 1
oneoffpurchasesfrequency	Purchase frequency in one-go, scored between 0 and 1
purchasesinstallments frequency	Installment purchase frequency, scored between 0 and 1
cashadvancefrequency	Cash in advance payment frequency, expressed as a score between 0 and 1
cashadvancetrx	Number of Transactions made with "Cash in Advanced"
purchases_trx	Number of purchase transactions made
credit_limit	Limit of Credit Card for user
payments	Amount of Payment done by user
minimum_payments	Minimum amount of payments made by user
prcfullpayment	Percent of full payment paid by user
tenure	Tenure of credit card service for user

To identify duplicates, we utilize the *cust_id* feature, deleting the latest occurrence in case of a duplicate. Missing values are handled through imputation with the median, chosen for its robustness to outliers and independence from the assumption of a normal data distribution commonly encountered in financial datasets. This approach substitutes each missing value with the median of the respective feature, maintaining the dataset's central tendency without being influenced by extreme values.

Following this, data normalization is performed to adjust values to a common scale without distorting differences in

[1] https://www.kaggle.com/datasets/arjunbhasin2013/c

ranges. The Standard Scaler method is employed, standardizing features by removing the mean and scaling to unit variance, which defined as Eq. (1):

$$z = \frac{x - \mu}{\sigma}, \tag{1}$$

where the new value z is calculated from the original value x, with μ as the mean of the feature and σ as the standard deviation.

B. Data Clustering

In this stage, our primary objective is to segment customers based on credit lending products, utilizing an improved k-means algorithm that incorporates PSO [9]. The integration of these algorithms aims to enhance the clustering process by leveraging the exploration and exploitation capabilities of PSO, while still benefiting from the efficiency of the k-means algorithm.

K-means classifies a dataset into a specified number of clusters (k) by minimizing the sum of distances between each data point and its cluster centroid. The algorithm initiates with the random assignment of k centroids as initial grouping centers. The iterative process involves assignment and update phases. In the assignment phase, each data point is allocated to the nearest centroid using Euclidean distance—a measure of the actual straight-line distance between two points in an n-dimensional space, defined as Eq. (2):

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}. \tag{2}$$

In this equation, x and y represent the data point and the centroid, respectively, and n signifies the number of dimensions. After that, in the update phase, centroids are recalculated by averaging points within each cluster. Iterations continue until centroids stabilize, indicating a locally optimal partition of data into clusters.

To improve the selection of initial centroids for k-means, PSO is applied. PSO iteratively improves candidate solutions by guiding particles in the search space based on a given measure of quality [10] [11]. It initializes particles, each with a random position c_i and velocity v_i in the search space. These particles move through the search space, influenced by their own best-known positions, called personal bests $p_{best,i}$, and the best-known position among all particles, known as the global best g_{best}. The movement of each particle is governed by updating its velocity, which reflects a balance between its own historical movement, the direction towards its personal best position, and the direction towards the global best position. The velocity update is defined as Eq. (3):

$$v_i^{t+1} = wv_i^t + c_1r_1\left(P_{best,i} - x_i^t\right) + c_2r_2(g_{best} - x_i^t) \tag{3}$$

where w is the inertia weight, and c_1 and c_2 are acceleration coefficients, and r_1 and r_2 are random numbers between 0 and 1. The particle's position is then updated by adding this new velocity to its current position.

The objective function optimized by PSO is the quality of clustering, measured by Within-Cluster Sum of Squares (WCSS), also known as the inertia. WCSS reflects the compactness of the clusters, with lower values indicating tighter clusters. The WCSS is calculated as the sum of the squared distances between each data point and the centroid of its assigned cluster. Mathematically, it can be expressed as Eq. (4):

$$WCSS = \sum_{i=1}^{k}\sum_{x\in C_i}\|x - \mu_i\|^2, \tag{4}$$

where k is the numbers of clusters, C_i is the set of points in the i-th cluster, x is a data point in clusters C_i, μ_i is the centroid of the i-th cluster, and $\|x - \mu_i\|^2$is the squared Euclidean distance between a data point x and the centroid μ_i of its clusters.

The optimal number of clusters for k-means is determined using the elbow method. K-Means is applied for various k values, and WCSS is calculated. The "elbow" point in the WCSS plot signifies a balance between minimizing WCSS and having a reasonable number of clusters [12]. The algorithm for k-means and the combination of k-means with PSO are presented in Fig. 1 and Fig. 2, respectively.

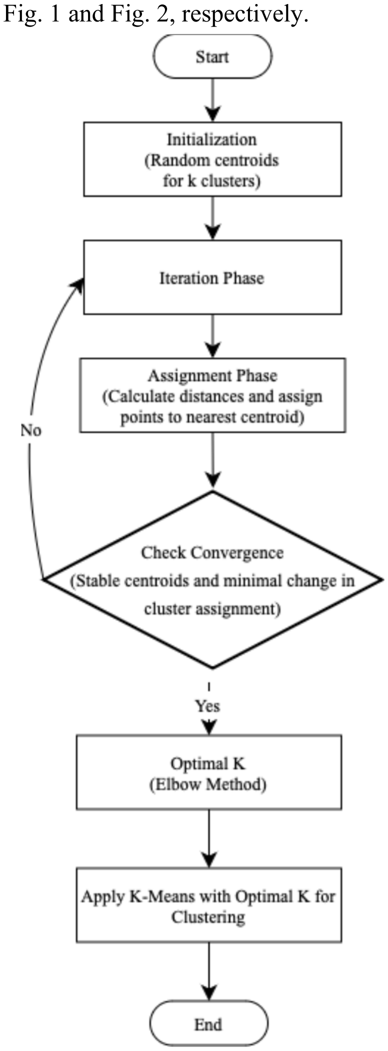

Fig. 1 Flowchart of k-means clustering.

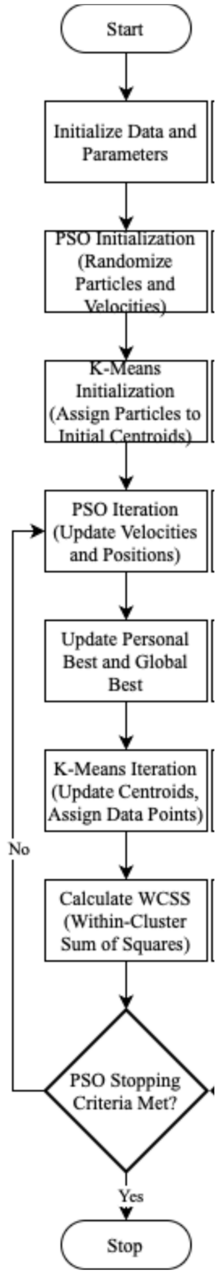

Fig. 2 The k-Means-PSO flowchart.

C. *Building Model*

Inspired by the capability of ANN for modeling complex relationships and patterns in data, we proposed a network architecture for predicting purchase value, illustrated in **Error! Reference source not found.**. The input layer of the network receives 16 features, as listed in Table I, excluding *cust_id* and *purchase* features. Cust_id is excluded as it is an identifier value that is not useful for predicting the purchase feature.

The number of neurons in the input layer corresponds to the number of features. The first hidden layer, comprising 64 neurons, initiates the process of extracting patterns and relationships from the data, utilizing the Rectified Linear Unit (ReLU) activation function. ReLU is chosen for its effectiveness in handling non-linear relationships, commonly employed in deep learning models. The number of neurons in each hidden layer can be adjusted based on the data complexity and desired level of abstraction.

The second hidden layer, with 32 neurons, further refines the learning process, distilling more abstract representations. The output layer, consisting of a single neuron, predicts the purchase value—a continuous variable—reflecting the nature of a regression task.

The model utilizes the Adam algorithm for optimization, employing mean squared error as the loss function [13]. Training occurs over 100 epochs, incorporating an early stopping mechanism that monitors the validation loss and terminates training if the model fails to improve after 10 epochs.

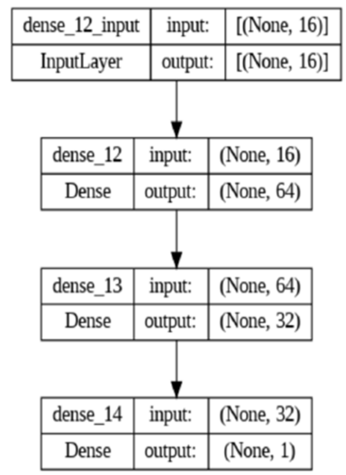

Fig. 3 Architecture of the proposed ANN.

V. EXPERIMENTS

In this section, we present a baseline implementation as a benchmark for evaluating the proposed approach, followed by a comprehensive comparative analysis of experimental results.

A. *Baseline*

We conduct a comparative analysis of the prediction performance between our proposed ANN and Support Vector Regression (SVR) [14], a type of Support Vector Machine (SVM) for regression tasks. In our study, we implement a non-linear SVR model that excels in discovering intricate patterns by employing kernel functions, transforming the data into a higher-dimensional space to capture non-linear dependencies.

To optimize the performance of the non-linear SVR model, we employ GridSearchCV. This approach systematically explores a range of hyperparameters, seeking the combination that yields the best predictive performance [15]. Through GridSearchCV, we evaluate various configurations of SVR's key parameters, including the regularization parameter (*C*), epsilon, and kernel. The parameters on our grid are detailed in Table II.

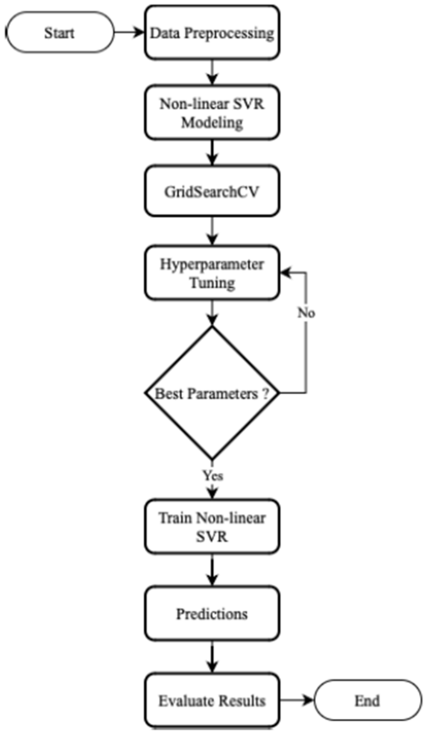

Fig. 4 Flowchart of SVR model use as a baseline method.

TABLE II
THE HYPER-PARAMETER SPACE OF SVR

Parameter	Value
C	0.1
	1
Epsilon	0.1
	0.001
Kernel	rbf
	sigmoid

Specifically, the regularization parameter, denoted as C, determines the trade-off between achieving a low error on the training data and minimizing the norm of the weights. In our grid, we test two values, 0.1 and 1, which are commonly used starting points in hyperparameter tuning. The epsilon parameter defines the margin of tolerance where errors are not penalized. Smaller values of epsilon imply a stricter fit to the training data. Our grid includes two values, 0.1 and 0.001, allowing us to explore the effects of a wider and a narrower margin. The kernel parameter specifies the type of kernel to be used. The 'rbf' kernel represents Radial Basis Function, a popular choice for non-linear data, capable of mapping an input space into an infinite-dimensional space. Meanwhile, the 'sigmoid' kernel provides a neural network-like interpretation. By including both 'rbf' and 'sigmoid', we compare a commonly used kernel for SVR with a less common but potentially useful alternative to understand how each affects the model's performance. After fitting 2 folds for each of 8 candidates on GridSearchCV, the optimal parameters obtained from the tuning are C equal to 1, epsilon equal to 0.1, and using the sigmoid kernel. The process is visually illustrated in Fig. 4.

B. *Results and Discussion*

This section presents the results of customer segmentation followed by an examination of purchase prediction.

1) Customer Segmentation: As previously described, the k-eans based model is employed for customer segmentation. Prior to segmentation, the optimal number of clusters is determined using the elbow method. As shown in Fig. 6. The elbow point, identified through this method, corresponds to 6 clusters.

The silhouette score is utilized to assess the performance of the k-means model, which is formulated as Eq. (5):

$$SilhouetteScore = \frac{b-a}{\max{(a,b)}}, \tag{5}$$

where a is the mean distance between the sample and all other points in the same cluster, and b is the smallest mean distance of the sample to all points in any other cluster, excluding the sample's own cluster. The silhouette score for the entire dataset is then calculated as the mean of the silhouette scores of each sample. Particularly, this metric gauges how similar an object is to its own cluster compared to other clusters, with a score ranging from -1 to 1. A high score indicates that an object is well-matched to its own cluster and poorly matched to neighboring clusters.

Table III presents the impact of employing PSO for optimizing the centroid selection. The silhouette score for original k-means alone is 0.2697, suggesting that, on average, the clusters are not very dense or well-separated. This may indicate some overlap or less homogeneity among the clusters. In contrast, k-means with PSO achieves a silhouette score of 0.2885, a slight improvement. This enhancement implies that the addition of PSO, a metaheuristic optimization technique, refines the clustering process, resulting in slightly more cohesive and better-separated clusters compared to k-means alone. PSO may have contributed to optimizing the centroid placement by considering a global best solution during centroid position updates, leading to better-defined clusters.

Although neither method produces highly dense clusters (indicated by silhouette scores less than 0.3), the use of PSO has a positive impact on the quality of customer segmentation. The scores suggest the possibility of significant overlap between segments or that the segments themselves are not particularly tight. For practical applications in customer segmentation, it may be necessary to explore other variables, consider a different number of clusters, or try alternative clustering algorithms to potentially yield more distinct and actionable segments.

TABLE III
SILHOUETTE SCORE OF DIFFERENT MODELS

Model	Silhouette Score
Original k-means (without PSO)	0.2697
Modified k-means (with PSO)	<u>0.2885</u>

2) Purchase Forecasting: To assess the effectiveness of the prediction model, we employ the Mean Squared Error (MSE) number and Root Mean Squared Error (RMSE). RMSE, being the square root of MSE, is measured in the same units as the target variable. MSE, measured in the square of the target variable units, penalizes larger errors more severely due to its formulation, akin to the squared loss function.

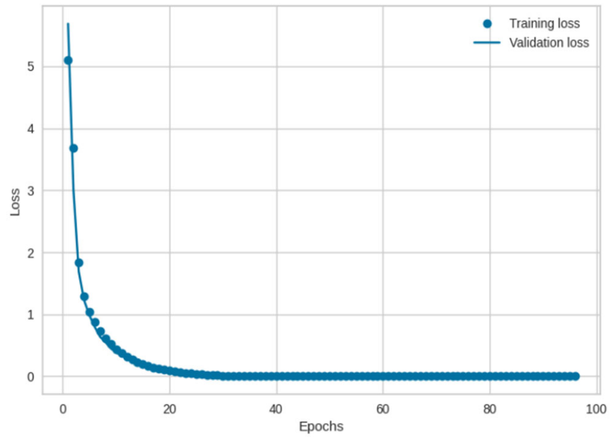

Fig. 5 Training and validation loss curves.

TABLE IV
SILHOUETTE SCORE OF DIFFERENT MODELS

Model	**Silhouette Score**
Original k-means (without PSO)	0.2697
Modified k-means (with PSO)	0.2885

Table IV provides a summary of the MSE and RMSE values for the proposed method in comparison to the baseline. The ANN model exhibits significantly lower error values for both MSE and RMSE, indicating superior performance on the given dataset. With an MSE of 284.742 and an RMSE of 16.874, the ANN model predictions closely align with the actual values. In contrast, the SVR model shows considerably higher errors with an MSE of 4,147,323.042 and an RMSE of 2,036.497. These elevated values suggest that the SVR model's predictions deviate significantly from the actual values, indicating a poorer fit to the data. The substantial difference in performance metrics suggests that the features and patterns in the dataset are more appropriately captured by the neural network's architecture than the hyperplane fitting approach used by SVR.

We provide a plot of training and validation loss learning curves in Fig. 5. The learning curve indicates that the model's training and validation losses decrease significantly and converge toward 0 after epoch 20. This convergence signifies the model's improving performance over time as it learns from the training data. The proximity of the validation loss to the training loss suggests that the model is not overfitting, a common concern in machine learning models. Overfitting occurs when a model learns the training data too well, capturing noise and details that do not generalize to unseen data. The decrease in both training and validation losses, approaching zero, implies that the model is effectively learning the underlying patterns and generalizing well to unseen data.

VI. CONCLUSION AND FUTURE WORKS

This study presents a comprehensive approach to customer segmentation and purchase forecasting, employing advanced techniques such as k-means clustering with PSO and an ANN for predictive modeling. The results showcase the effectiveness of the proposed methodologies, particularly in customer segmentation, where the addition of PSO refines clustering, leading to better-defined segments. Additionally, the ANN model outperforms SVR in purchase forecasting, demonstrating significantly lower error values and a more robust fit to the dataset. The learning curves further affirm the ANN model's capability to learn and generalize patterns effectively. These findings underscore the importance of leveraging advanced algorithms for data processing, clustering, and predictive modeling in enhancing the precision of customer-related tasks. Future research could explore further optimizations and alternative methodologies to continually advance the accuracy and applicability of such models in real-world business scenarios.

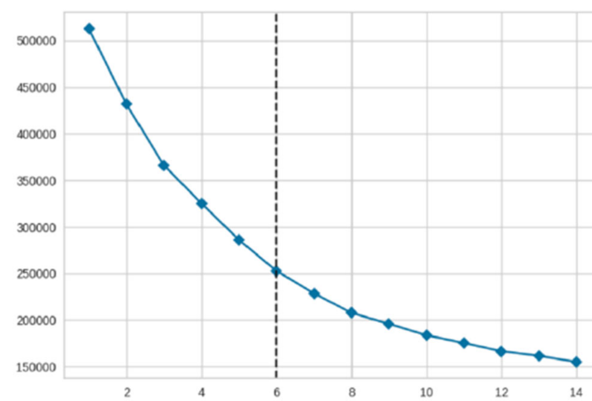

Fig. 6 Finding the optimal number of clusters using the elbow method.

REFERENCES

[1] M. H. U. Rashid, M. Nurunnabi, M. Rahman and M. A. K. Masud, "Exploring the Relationship between Customer Loyalty and Financial Performance of Banks: Customer Open Innovation Perspective," *Journal of Open Innovation: Technology, Market, and Complexity,* vol. 6, no. 4, 2020.

[2] M. Ala'raj, M. F. Abbod and M. Majdalawieh, "Modelling customers credit card behaviour using bidirectional LSTM neural networks," *Journal of Big Data,* vol. 8, no. 69, 2021.

[3] S. Arora, S. Bindra, S. Singh and V. K. Nassa, "Prediction of credit card defaults through data analysis and machine learning techniques," *Materialstoday: Proceedings,* vol. 51, pp. 110-117, 2022.

[4] Z. Xueping, L. Qingnian, H. Ying, H. Lei and D. Pengying, "Fraud Prediction of Credit Card Customers Based on Xgboost Model and Multi-Layer Perception Model," in *IEEE International Conference on Advances in Electrical Engineering and Computer Applications*, 2022.

[5] K.-L. Hua, C.-H. Hsu, S. C. Hidayati, W.-H. Cheng and Y.-J. Chen, "Computer-aided classification of lung nodules on computed tomography images via deep learning technique," *OncoTargets and therapy,* vol. 8, pp. 2015-2022, 2015.

[6] V. N. Dornadula and S. Geetha, "Credit Card Fraud Detection using Machine Learning Algorithms," *Procedia Computer Science,* vol. 165, pp. 631-641, 2019.

[7] R. Asha and K. S. Kumar, "Credit card fraud detection using artificial neural network," *Global Transitions Proceedings,* vol. 2, no. 1, pp. 35-41, 2021.

[8] C.-F. Wu, S.-C. Huang, C.-C. Chiou and Y.-M. Wang, "A predictive intelligence system of credit scoring based on deep multiple kernel learning," *Applied Soft Computing,* vol. 111, no. 21, p. 107668, 2021.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995.

[10] A. Ahmadyfard and H. Modares, "Combining PSO and k-means to enhance data clustering," in *International Symposium on Telecommunications*, 2008.

[11] O. Irnawati and N. Nuraeni, "Optimasi Decision Tree Dengan Particle Swarm Optimization Untuk Klasifikasi Prediksi Gender Pengguna E-Commerce," *Journal of Information Engineering and Educational Technology,* vol. 6, no. 1, pp. 37-41, 2022.

[12] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman and B. D. Satoto, "Integration K-Means Clustering Method and Elbow Method for Identification of the Best Customer Profile Cluster," in *IOP Conference Series: Materials Science and Engineering*, 2018.

[13] D. P. Kingma och J. Ba, "Adam: A Method for Stochastic Optimization," i *Proc. International Conference on Learning Representations*, 2015.

[14] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola and V. Vapnik, "Support vector regression machines," in *International Conference on Neural Information Processing Systems*, 1996.

[15] A. W. Ishlah, S. Sudarno and P. Kartikasari, "Implementasi GridSearchCV pada Support Vector Regression," *Jurnal Gaussian,* vol. 12, no. 2, pp. 276-286, 2023.

[16] M. Ahmed, R. Seraj and S. M. S. Islam, "The kmeans algorithm: A comprehensive survey and performance evaluation," *Electronics,* pp. 1-12, 2020.