Revue d'Intelligence Artificielle
Vol. 38, No. 4, August, 2024, pp. 1223-1233
Journal homepage: http://iieta.org/journals/ria

# Optimizing the Evaluation of K-means Clustering Using the Weight Product

Rozzi Kesuma Dinata*, Bustami, Sujacka Retno

Department of Informatics, Universitas Malikussaleh, Aceh 24353, Indonesia

Corresponding Author Email: rozzi@unimal.ac.id

https://doi.org/10.18280/ria.380416

**Received:** 26 October 2023
**Revised:** 12 March 2024
**Accepted:** 26 June 2024
**Available online:** 23 August 2024

***Keywords:***
*K-means, clustering, Weight Product Model (WPM), iterations, evaluation*

## ABSTRACT

In the process of the K-means clustering algorithm, one of the issues that arises is the high number of iterations. This study aims to optimize the cluster evaluation results in K-means by reducing iterations through the application of the Weight Product Model (WPM). The evaluation method used in this research is the Davies-Bouldin Index (DBI). Three datasets were analyzed: the QSAR Dataset consisting of 908 data points, 7 attributes; the Whoscale Customer dataset consisting of 440 data points, 8 attributes from the UCI Machine Learning Repository, as well as direct observational data from captured fisheries obtained from the North Aceh District Office of Marine and Fisheries, Indonesia consisting of 75 data points, 8 attributes. The results of 10 testing iterations on three different datasets show that for the QSAR Dataset, the average cluster evaluation using DBI with K-means is 0.852. However, when applying WPM+K-means, the average DBI value increases to 0.727, with the average number of K-means iterations reduced from 23 to 8 iterations. For the Whoscale Customer dataset, the average cluster evaluation using DBI with K-means is 0.921. In contrast, when employing WPM+K-means, the average DBI value slightly improves to 0.910, accompanied by a reduction in the average number of K-means iterations from 23 to 10 iterations. In the case of the captured fisheries dataset, the average cluster evaluation using DBI with K-means yields a value of 1.222. However, implementing WPM+K-means results in an improved average DBI of 1.052. Furthermore, the average number of K-means iterations is reduced to 9 iterations, whereas for WPM+K-means, this number is reduced to 4 iterations. The results of this study demonstrate an improvement in DBI values, where lower DBI values indicate better performance of the K-means algorithm. These also findings demonstrate that WPM is effective in optimizing cluster evaluation values in K-means clustering. With the reduction in the number of K-means iterations, computational time is expected to be faster.

## 1. INTRODUCTION

The K-means algorithm is a commonly used clustering technique in the field of data science. Its primary objective is to divide a dataset into a predetermined number of clusters based on shared features or characteristics [1]. However, the K-means algorithm has limitations, including sensitivity to initial centroid placement, which can lead to convergence on local minima. Additionally, it tends to produce uniform clusters, limiting its adaptability to varied cluster shapes and sizes. Outliers can distort centroids, resulting in inaccurate outcomes. The algorithm assumes spherical and equal clusters, making it less effective for complex and diverse structures. Scaling also affects its performance, emphasizing features with varying scales. Despite its popularity, K-means may struggle with intricate cluster geometries [2]. The abundance of iteration processes contributes to suboptimal clustering performance [3]. One technique for evaluating K-means performance involves the use of clustering evaluation methods [4], such as the Silhouette Score, Davies-Bouldin Index (DBI), Calinski-Harabasz Index (Variance Ratio Criterion), and Inertia (Within-Cluster Sum of Squares) [5]. The DBI quantitatively measures clustering quality by calculating the average similarity between each cluster and its most similar neighboring cluster, relative to its internal similarity. A lower DBI value indicates better clustering, highlighting clusters that are internally cohesive and well-separated. DBI is advantageous for its ability to provide a single numerical value summarizing clustering performance, enabling easy comparison across different algorithms [6]. Conversely, the Silhouette Score assesses cluster cohesion and separation within a dataset. It quantifies the similarity of each data point to its assigned cluster compared to neighboring clusters. A score close to +1 suggests effective clustering, while a score near -1 indicates potential misassignment. The Silhouette Score offers insights into cluster compactness and isolation [7].

Numerous scholars have extensively researched K-means clustering, resulting in contributions from various researchers. For instance, Sinaga and Yang [8] investigated a groundbreaking unsupervised K-means (U-K-means) clustering algorithm and thoroughly analyzed its computational complexity. Ikotun et al. [9] highlighted the limitation of the algorithm's reliance on Euclidean distance as a similarity measure, hindering its ability to identify diverse cluster shapes and manage overlapping clusters. Ahmed et al. [10] discussed various K-means algorithm variants and their recent developments, assessing their effectiveness through experimental analyses on diverse datasets. Govender and Sivakumar [11] conducted a review encompassing 100 research articles from 1980 to 2019, focusing on the K-means method's utilization and the frequent application of average and Ward linkages in hierarchical clustering. Uddin and Roy [12] used clustering techniques to identify metro stations suitable for transit-oriented development (TOD) in Dhaka. They evaluated 17 stations on MRT line 6 based on nine characteristics, resulting in five clusters of stations with related features. Zhuang et al. [13] discussed the use of the commonly adopted K-means method, which partitions data into fewer groups using a distance-based approach. Rezaee et al. [14] explored the game-based K-means (GBK-means) algorithm, where cluster centers compete for similar data points to minimize distances from the maximum number of data points within their respective clusters.

Nguyen et al. [15] introduced a novel extension of the K-means method tailored for clustering categorical data. Aldino et al. [16] applied the K-means clustering method to two years of corn crop data to gain insights into the feasibility of corn production across different sub-districts. Barile et al. [17] employed the K-means clustering algorithm to analyze selected features and establish relationships, revealing a strong correlation between the amplitude of AE signals and the Frequency Centroid (C-Freq). Dinata et al. [18] proposed the utilization of the K-means algorithm to cluster data related to the number of regions and plant types in East Aceh Regency, sourced from the Department of Agriculture, Food Crops and Horticulture, East Aceh Regency. Rengasamy and Murugesan [19] focused on the Integrated K-means Laplacian (IKL) algorithm, incorporating attribute information and pairwise relational data for clustering. The IKL algorithm faces challenges in constructing the normalized Laplacian matrix, prompting the introduction of methodologies to enhance matrix creation and leading to the formulation of three new iterations of the IKL algorithm.

Based on previous research, this study proposes the utilization of the Weight Product (WP) method for selecting initial centroids in the K-means clustering algorithm. The Weight Product method, within the framework of fuzziness, calculates weights or importance of individual variables within a fuzzy-based system, considering membership degrees of variables to linguistic sets. WP's ranking outcomes determine initial centroids for K-means, aiming to optimize clustering outcomes. To evaluate K-means performance, this research employs DBI method, a metric crucial in clustering evaluation within data analysis. Lower DBI values indicate more effective cluster partitioning. Furthermore, this research conducts a comparative analysis between iterations of the K-means algorithm and WP + K-means, including the computation of DBI values for each test iteration.

## 2. DATASET

Within this study, three distinct datasets were utilized for the purpose of evaluating the proposed methodology. These datasets encompass the QSAR Dataset and Whoscale Customer Dataset, both acquired from the UCI Machine Learning Repository. Additionally, direct observational data regarding captured fisheries were sourced from the North Aceh District Office of Marine and Fisheries, Indonesia. A summary of the overall details pertaining to these datasets is presented in Table 1.

**Table 1.** The general dataset details

| Dataset | Number of Attributes | Number of Data Points |
|---|---|---|
| QSAR | 7 | 908 |
| Whoscale Customer | 8 | 440 |
| Captured Fisheries | 8 | 75 |

The selected datasets for this study were chosen due to their diverse attributes and data points, with the aim of conducting a thorough evaluation of the effectiveness of the proposed Weight Product Model (WPM) method in optimizing K-means clustering outcomes. The QSAR Dataset includes 8 attributes with 908 data points, the Whoscale Customer dataset consists of 8 attributes with 440 data points, and the Captured Fisheries Dataset covers 8 attributes with 75 data points. This deliberate assortment aims to offer a robust testing environment for evaluating the method's performance across different dataset attributes. The diverse attributes and data structures of these datasets allow for a comprehensive assessment of the proposed technique's performance across various domains. This demonstrates the adaptability and robustness of the proposed method in handling diverse data types and complexities.

## 3. THE PROPOSED ALGORITHM

### 3.1 K-means clustering

The K-means algorithm is utilized for data clustering. The steps for clustering using K-means are as follows. In Step 1, the number of clusters to be created, denoted as 'k' is established. In Step 2, initial random values are assigned to each centroid of the 'k' clusters. The distance between each data point and the centroids is calculated using the Euclidean distance formula [20]:

$$d(xi,\mu j)=\sqrt{\sum(xi-\mu j)^2} \tag{1}$$

where, $d$ represents the data point, $xi$ stands for the data criteria, and $\mu j$ represents the centroid of cluster $j$. Moving on to Step 3, each data point is grouped based on its proximity to the nearest centroid. In Step 4, the centroids are updated by computing the average of the data within each cluster using the formula:

$$\mu j(t+1)=\frac{1}{Nsj}\sum_{j\in sj} xj \tag{2}$$

In this context, the symbol $\mu j(t+1)$" signifies the centroid that has been updated during the iteration $(t+1)$, which reflects the evolving central point of a distinct cluster. The term $Nsj$ corresponds to the dataset residing within the cluster denoted as $Sj$, thereby indicating the compilation of data points grouped within that specific cluster. Furthermore, "xj" denotes the accumulation of values encompassed by the cluster Sj, efficiently encapsulating the cumulative attributes of the clustered data points.

Finally, Step 5 concludes the process. Steps 2 to 4 are reiterated until there are no further alterations in the membership of each cluster, demonstrating the convergence of outcomes. This indicates that the algorithm has successfully achieved consistent cluster assignments.

### 3.2 Weight Product Model (WPM)

WPM is an extension of the Weighted Sum Model (WSM). In the WPM framework, each alternative is systematically compared with other alternatives by utilizing multiplication of distinct ratios. These ratios correspond to specific decision criteria. Each ratio is raised to the power equivalent to the weight assigned to the corresponding criterion. The overarching formula for computing the Weighted Product (WP) score for alternative *jj* is expressed as follows [21]:

$$WP_j = \prod_{i=1}^{n} x_{ij}^{w_i} \tag{3}$$

In this specific context, the subsequent notations will be engaged: *n* will serve to denote the count of criteria being subjected to evaluation, w will be representative of the number of alternatives under contemplation, *wi* will be designated to stand for the weight allocated to criterion *ij* (where *i*=1,2,…,*ni*=1,2,…,*n*), and *xij* will indicate the standardized value corresponding to criterion ii for alternative *jj* (where *i*=1,2,…,*ni*=1,2,…,*n* and *j*=1,2,…,*mj*=1,2,…,*m*).

Contained within this formulation, the symbol ∏ will be enlisted to symbolize the product representation, signifying the multiplication of all criterion values raised to their corresponding criterion weights.

In WPM, the assignment of weights to each criterion is guided by their respective significance in the decision-making process. These weights signify the impact of individual criteria on the ultimate evaluation.

### 3.3 Davies Bouldin Index (DBI)

The DBI assesses clusters within clustering by considering both cohesiveness and separation aspects. Cohesion gauges the proximity of data to a cluster's centroid, while separation evaluates the closeness between centroids of different clusters. The process of calculating the Davies-Bouldin Index involves the following steps. Firstly, compute the Sum of Squares Within Cluster (SSW) to measure cohesion [22]:

$$SSWi = \frac{1}{mi} \sum_{j=i}^{mi} d(xj, ci) \tag{4}$$

This aspect gauges the cohesion of clusters by examining the closeness of data points within each cluster to their respective centroid. A lower SSW value indicates that the data points within each cluster are closer to their centroid, indicating higher cohesion within the cluster. Essentially, SSW reflects the degree of compactness or tight clustering of data points around their centroid in each cluster. Secondly, determine the Sum of Squares Between Cluster (SSB) to quantify separation:

$$SSBi, j = d(ci, cj) \tag{5}$$

Conversely, SSB evaluates the separation between clusters by measuring the distance between centroids of different clusters. A larger SSB value implies greater separation between clusters, suggesting that the centroids are more distant from each other. Conceptually, SSB assesses how distinct or well-separated the clusters are from each other in the feature space. Thirdly, ascertain the Ratio to compare clusters *ii* and *jj*:

$$Rij = \frac{SSWi + SSWj}{SSBij} \tag{6}$$

Lastly, calculate the DBI by employing the aforementioned ratios:

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} \left( R_{i,j} \right) \tag{7}$$

Diminished DBI values signify more well-structured clusters resulting from the clustering process.

## 4. PROPOSED MODEL

In this study, we propose integrating the WPM with K-means clustering. WPM is utilized to optimize the evaluation scores of K-means by determining initial centroids. Following this, the process transitions to the K-means clustering phase. Additionally, the study compares the performance of conventional K-means against WPM + K-means by calculating and contrasting DBI values across three datasets: QSAR, Whoscale Customer, and Captured Fisheries. Lower DBI values indicate more optimal clustering performance. We also analyze the number of iterations and the resulting clusters produced by both WPM K-means and conventional K-means. To provide further clarity, the structure of the research is depicted in Figure 1.

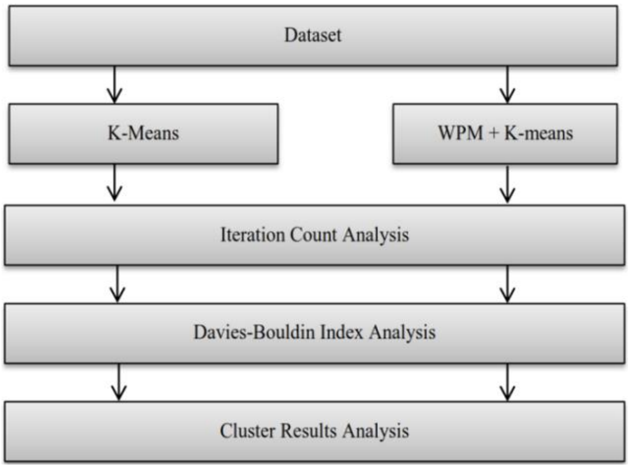

**Figure 1.** The proposed structure

## 5. RESULTS AND DISCUSSIONS

### 5.1 QSAR dataset

5.1.1 K-means

In this study, we conducted 10 tests. Here are the results from the first K-means analysis on the QSAR dataset. Initially, we randomly selected centroids as shown in Table 2. Next, we calculated the Euclidean distances between data points, with results presented in Table 3.

D900=$\sqrt{(3,26-2986)^2+(0,829-0,961)^2+(1676-1669)^2+(0-0)^2+(1-4)^2+(1453-1798)^2+(3770-3152)^2}$=3065,57

D907=$\sqrt{(3,26-2831)^2+(0,829-1393)^2+(1676-1077)^2+(0-0)^2+(1-1)^2+(1453-0,906)^2+(3770-5317)^2}$=3846,41

D908=$\sqrt{(3,26-4057)^2+(0,829-1032)^2+(1676-1183)^2+(0-1)^2+(1-3)^2+(1453-4754)^2+(3770-8201)^2}$=6947,63

**Table 2.** Initial centroids in the QSAR dataset for K-means

| Test Number | Test Data |
|---|---|
| 1 | 900, 907, 908 |
| 2 | 789, 799, 873 |
| 3 | 4, 7, 350 |
| 4 | 332, 374, 576 |
| 5 | 62, 285, 893 |
| 6 | 894, 896, 908 |
| 7 | 282, 647, 905 |
| 8 | 662, 880, 900 |
| 9 | 796, 865, 907 |
| 10 | 841, 897, 901 |

**Table 3.** K-means distance calculations for QSAR dataset

| No | C1 | C2 | C3 | Cluster |
|---|---|---|---|---|
| 1 | 3065,57 | 3846,41 | 6947,63 | 1 |
| 2 | 1903,10 | 3189,44 | 6589,31 | 1 |
| 3 | 1967,30 | 2932,53 | 6293,14 | 1 |
| 4 | 410,79 | 2942,99 | 5735,77 | 1 |
| 5 | 2931,68 | 2682,97 | 4737,22 | 2 |
| .. | …. | …. | …. | …. |
| .. | …. | …. | …. | …. |
| 20 | 1894,37 | 2939,20 | 7420,39 | 1 |
| 21 | 2295,64 | 3524,73 | 7995,44 | 1 |
| 22 | 1023,94 | 3445,11 | 6979,88 | 1 |
| 23 | 3629,88 | 5543,37 | 9610,03 | 1 |
| .. | …. | …. | …. | …. |
| .. | …. | …. | …. | …. |
| 907 | 3198,29 | 0,00 | 5705,61 | 2 |
| 908 | 6056,12 | 5705,61 | 0,00 | 3 |

The third step involves computing the new centroid for Cluster 1 by summing the values of each attribute and then dividing the sum by the total number of data points in Cluster 1 for each attribute. For instance, Attribute 1 is calculated as 1636640/668, yielding a value of 2450.06. Similarly, Attribute 2 can be determined as 59093.6/639, resulting in a value of 92.4783. Attribute 3 is obtained by dividing 645791 by 668, giving a value of 966.752. Likewise, Attribute 4 is computed as 124/92, resulting in a value of 1.34783. Attribute 5 yields a value of 1.41451 when 273 is divided by 193. Attribute 6 is calculated as 1183275/668, resulting in a value of 1771.37. Lastly, Attribute 7 is found by dividing 2341868 by 668, yielding a value of 3505.79. Next, calculate the new centroid for Cluster 2 by summing the values of each attribute and dividing by the total number of data points in Cluster 2 for each attribute. Following that, calculate the new centroid for Cluster 2 by summing the values of each attribute and dividing by the total number of data points in Cluster 2 for each attribute. The results of the new centroids in the first iteration of K-means on the QSAR Dataset are presented in Table 4. Repeat all these steps until the final centroid converges with the previous one. In the first test, K-means stopped at the 20th iteration, which is displayed in Table 5. Table 5 indicates that the new centroid values in the 20th iteration have converged with the centroid values from the 19th iteration, leading to the termination of the K-means process. The results of the first test show that data point 900 is in Cluster 1, data point 907 is also in Cluster 1, and data point 908 belongs to Cluster 3.

### 5.1.2 WPM + K-means

In this research, the first step in determining the WPM value involves establishing the weight for each attribute. In the QSAR dataset, the maximum weight is set at 1. Next, calculate 1/7 per attribute, resulting in a weight of 0.143 per attribute.

The second step is to calculate the maximum value for each attribute. The third step involves dividing data 1 by the maximum value of attribute 1, then multiplying it by the weight of attribute 1. Next, add the value of data 2 divided by the maximum value of attribute 2, and multiply it by the weight of attribute 2, as shown below. The WPM calculation result on the QSAR dataset is shown in Table 6, and the initial centroids used on the QSAR dataset for WPM + K-means are shown in Table 7.

**Table 4.** The results of the first iteration of K-means testing on the QSAR dataset

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **C1** | 2450,061 | 92,478 | 966,752 | 1,348 | 1,415 | 1771,370 | 3505,791 |
| **C2** | 2139,303 | 483,801 | 870,224 | 1,469 | 1,750 | 579,079 | 4924,215 |
| **C3** | 3298,576 | 713,919 | 680,832 | 1,542 | 1,867 | 4251,109 | 6364,082 |

**Table 5.** The results of the 20th iteration of K-means testing on the QSAR dataset

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **C1** | 2615,378 | 79,942 | 1346,201 | 1,156 | 1,473 | 506,400 | 2835,344 |
| **C2** | 1249,445 | 224,178 | 556,195 | 1,234 | 1,421 | 1913,627 | 4277,355 |
| **C3** | 3354,159 | 386,287 | 757,155 | 1,750 | 1,664 | 3311,824 | 5128,783 |

Data point 3:
=(13,53/146,28)*0,143+(22,50/137,74)*0,143+(66,91/452,15)*0,143+(20,91/122,61)*0,143+(34,69/303,66)*0,143+(8,49/401,26)*0,143+(34,69/258,03)*0,143+(14,64/174,83)*0,143=0,183
Data point 19:
=(7,56/146,28)*0,143+(9,76/137,74)*0,143+(45,14/452,15)*0,143+(17,20/122,61)*0,143+(12,32/303,66)*0,143+(1/401,26)*0,143+(6,47/258,03)*0,143+(7,56/174,83)*0,143=0,289
Data point 23:
=(39,98/146,28)*0,143+(43,05/137,74)*0,143+(49,69/452,15)*0,143+(44,53/122,61)*0,143+(32,60/303,66)*0,143+(32,47/401,26)*0,143+(31,98/258,03)*0,143+(61,38/174,83)*0,143=0,340

**Table 6.** WPM calculation result on the QSAR dataset

| NO | CIC0 | SM1_Dz(Z) | GATS1i | NdsCH | NdssC | MLOGP | Quantitative Response, LC50 [-LOG (mol/L)] | WPM |
|---|---|---|---|---|---|---|---|---|
| 1 | 3,26 | 0,829 | 1.676 | 0 | 1 | 1.453 | 3.770 | 0,250428 |
| 2 | 2.189 | 0,58 | 0,863 | 0 | 0 | 1.348 | 3.115 | 0,176684 |
| 3 | 2.125 | 0,638 | 0,831 | 0 | 0 | 1.348 | 3.531 | 0,183862 |
| .. | … | …. | … | …. | … | … | …. | …… |
| .. | … | …. | … | …. | … | … | …. | …… |
| 827 | 2,43 | 0,496 | 0,83 | 1 | 0 | 1.132 | 4.628 | 0,17104 |
| 828 | 2.435 | 1.113 | 1.109 | 0 | 1 | 2.354 | 2.839 | 0,37943 |
| 829 | 3.247 | 0,874 | 1.221 | 0 | 1 | 2.659 | 4.262 | 0,370646 |
| .. | … | …. | … | …. | … | … | …. | …… |
| .. | … | …. | … | …. | … | … | …. | …… |
| 834 | 2.233 | 0,57 | 0,883 | 0 | 0 | 1.501 | 3.325 | 0,187156 |
| 835 | 3.179 | 0 | 1.063 | 0 | 0 | 2.942 | 3.811 | 0,333395 |
| .. | … | …. | … | …. | … | … | …. | …… |
| .. | … | …. | … | …. | … | … | …. | …… |
| 906 | 3.763 | 0,916 | 0,878 | 0 | 6 | 2.918 | 4.818 | 0,452388 |
| 907 | 2.831 | 1.393 | 1.077 | 0 | 1 | 0,906 | 5.317 | 0,39648 |
| 908 | 4.057 | 1.032 | 1.183 | 1 | 3 | 4.754 | 8.201 | 0,700479 |

**Table 7.** Initial centroids in the QSAR dataset for WPM + K-means

| Test Number | Test Data |
|---|---|
| 1 | 96, 268, 493 |
| 2 | 468, 724, 772 |
| 3 | 83, 235, 847 |
| 4 | 285, 556, 747 |
| 5 | 189, 214, 766 |
| 6 | 131, 205, 311 |
| 7 | 6, 62, 73 |
| 8 | 23, 293, 561 |
| 9 | 96, 147, 883 |
| 10 | 716, 827, 893 |

**Table 8.** The results of the WPM + K-means distance calculations on the QSAR dataset

| No | C1 | C2 | C3 | Cluster |
|---|---|---|---|---|
| 1 | 3076,64 | 7183,33 | 2863,62 | 3 |
| 2 | 2422,01 | 6724,75 | 1163,64 | 3 |
| 3 | 2552,80 | 6477,41 | 903,23 | 3 |
| .. | …. | …. | …. | …. |
| .. | …. | …. | …. | …. |
| 20 | 778,36 | 7999,65 | 2835,83 | 1 |
| 21 | 966,57 | 8402,31 | 3015,03 | 1 |
| 22 | 1412,50 | 7342,44 | 2332,58 | 1 |
| 23 | 2651,56 | 9927,30 | 4914,84 | 1 |
| .. | …. | …. | …. | …. |
| .. | …. | …. | …. | …. |
| 907 | 3209,08 | 6624,61 | 3128,21 | 3 |
| 908 | 7643,62 | 2162,03 | 5529,65 | 2 |

Using the WPM results, the initial centroids can be determined, and this process will be repeated ten times for testing. In the first test, the initial centroids are selected from data point 96, data point 268, and data point 493. After determining the initial centroids obtained from the WPM calculation, the next step is to calculate the cluster distances using the Euclidean distance method, as shown in Table 8. The next step is to calculate the new C1 centroid by computing the average of each attribute's values for all data points assigned to C1, which is achieved by summing up the values of each attribute and dividing by the total number of data points assigned to C1 for each attribute.

In order to calculate the new centroid for a specific cluster, we employed a straightforward method. We divided the sum of each attribute by the total number of data points contained within that cluster for that particular attribute. The resulting averages are as follows: For Attribute 1, the average value is 2372.935. This was determined by dividing the sum of 661682.715 by the total of 279 data points in the cluster. Attribute 2 has an average of 98.478, calculated by dividing the sum of 27371.938 by the 278 data points in the cluster. Attribute 3's average value is 1510.265, which was derived from dividing the sum of 421468.501 by the 279 data points in the cluster. Moving on to Attribute 4, it has an average of 1.21875, calculated by dividing 39 by the 32 data points within the cluster. For Attribute 5, the average value is 1.48936, obtained by dividing 140 by the 94 data points in the cluster. Attribute 6's average is 262.027, resulting from dividing the sum of 73064.424 by the 279 data points in the cluster. Lastly, Attribute 7 has an average of 2746.81, which was calculated by dividing the total of 765412.446 by the 279 data points within the cluster.

The average value for C2 is obtained by dividing the total attribute sum for C2 (281554.29) by the number of data points assigned to C2 (79), resulting in an average of approximately 3562.430. To calculate the C2 average, the total attribute sum for C2 (60031.427) is divided by the number of data points assigned to C2 (78), yielding an average value of approximately 769.885. By dividing the total attribute sum for C2 (56314.429) by the number of data points assigned to C2 (79), an average value of roughly 713.566 is obtained. The average value for C2 is found by dividing the total attribute sum for C2 (30) by the number of data points assigned to C2 (19), resulting in an average value of 1.57895. Similarly, the average value for C2 is calculated by dividing the total attribute sum for C2 (69) by the number of data points assigned to C2 (35), resulting in an average of approximately 1.97143. The C2 average is determined by dividing the total attribute sum for C2 (356295) by the number of data points assigned to C2 (79), yielding an average value of roughly 4503.291. Lastly, to find the average value for C2, the total attribute sum for C2 (512747) is divided by the number of data points assigned to C2 (79), resulting in an average value of approximately 6484.544.

To calculate the new centroid for C3, we followed a straightforward method. We divided the total sum of each attribute by the number of data points specifically in C3 for that attribute. The results are as follows: For the first attribute, the average value is 2426.012, obtained by dividing the sum of

1334356.257 by the total of 550 data points in C3. The second attribute's average is 210.75, derived from dividing the sum of 108808.447 by the 516 data points in C3. The third attribute has an average of 647.323, calculated by dividing the total of 356028.181 by the 550 data points in C3. Moving on to the fourth attribute, it has an average of 1.43299, found by dividing 139 by the 97 data points within C3. For the fifth attribute, the average value is 1.47771, obtained by dividing 232 by the 157 data points in C3. The sixth attribute's average is 2357.85, calculated by dividing the sum of 1296817.819 by the 550 data points in C3. Lastly, the seventh attribute's average is 4370.738, resulting from dividing the total of 2403906 by the 550 data points within C3. The results of the new centroids in the first iteration of WPM + K-means on the QSAR dataset are presented in Table 9. Repeat all these steps until the final centroid converges with the previous one. In the first test, WPM + K-means stopped at the 7th iteration, which is displayed in Table 10. The results of the first test show that data point 96 is in Cluster 3, data point 268 is in Cluster 1, and data point 493 belongs to Cluster 2. The WPM + K-means method has demonstrated that the number of iterations required in K-means is significantly reduced compared to conventional K-means.

5.1.3 Comparison of DBI values on the QSAR dataset

The results of the iteration comparison across ten different tests between conventional K-means and WPM + K-means are presented in Table 11. The comparison of the number of iterations for K-means and WPM + K-means on the QSAR dataset is displayed in Figure 2, while the comparison of the DBI values between WPM + K-means and conventional K-means on the same dataset is shown in Figure 3. The clustering results are depicted in Figures 4 and 5.

**Table 9.** The results of the first iteration of WPM + K-means testing on the QSAR dataset

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **C1** | 2450,061 | 92,478 | 966,752 | 1,348 | 1,415 | 1771,370 | 3505,791 |
| **C2** | 2139,303 | 483,801 | 870,224 | 1,469 | 1,750 | 579,079 | 4924,215 |
| **C3** | 3298,576 | 713,919 | 680,832 | 1,542 | 1,867 | 4251,109 | 6364,082 |

**Table 10.** The results of the 7th iteration of WPM+K-means testing on the QSAR dataset

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **C1** | 2615,378 | 79,942 | 1346,201 | 1,156 | 1,473 | 506,400 | 2835,344 |
| **C2** | 1249,445 | 224,178 | 556,195 | 1,234 | 1,421 | 1913,627 | 4277,355 |
| **C3** | 3354,159 | 386,287 | 757,155 | 1,750 | 1,664 | 3311,824 | 5128,783 |

**Table 11.** The comparison of the number of iterations and DBI Value on the QSAR dataset

| Test Data | K-means Iterations | WPM+K-means Iterations | K-means DBI Value | WPM+K-means DBI Value |
|---|---|---|---|---|
| 1 | 20 | 7 | 0,678 | 0,514 |
| 2 | 22 | 9 | 0,667 | 0,538 |
| 3 | 20 | 8 | 0,917 | 1,637 |
| 4 | 32 | 10 | 0,610 | 0,599 |
| 5 | 16 | 7 | 0,750 | 0,618 |
| 6 | 21 | 10 | 0,910 | 0,682 |
| 7 | 18 | 8 | 0,703 | 0,751 |
| 8 | 27 | 9 | 0,566 | 0,587 |
| 9 | 35 | 10 | 1,775 | 0,771 |
| 10 | 22 | 7 | 0,942 | 0,571 |
| Average | 23.3 | 8.5 | 0,852 | 0,727 |

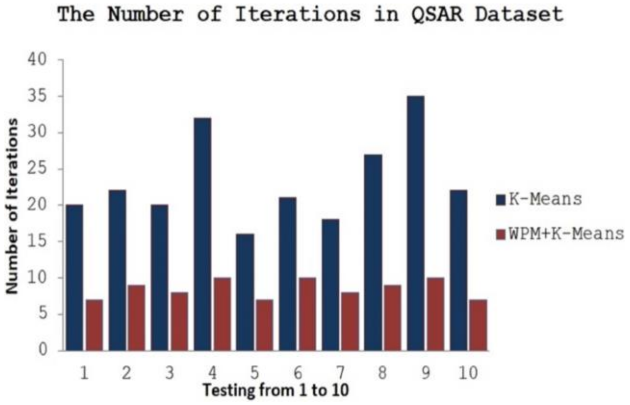

**Figure 2.** The comparison of the number of iterations

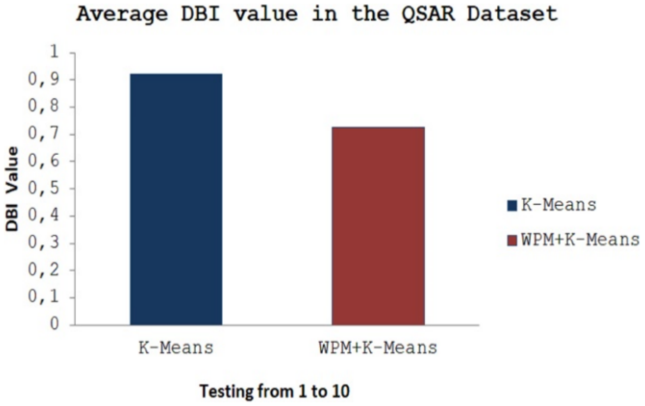

**Figure 3.** The comparison of DBI value

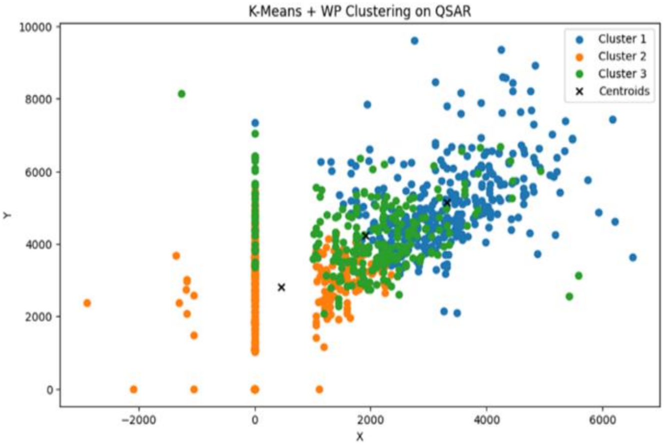

**Figure 4.** The clustering results on WPM + K-means on QSAR dataset

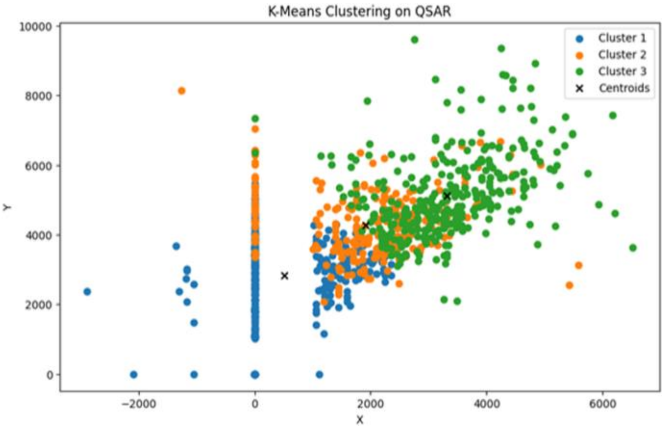

**Figure 5.** The clustering results on K-means on QSAR dataset

## 5.2 Whoscale Customer dataset

### 5.2.1 K-means

Conventional K-means computations were executed on the Whoscale Customer dataset, with a total of 10 testing iterations utilizing distinct randomly selected test datasets. The test datasets employed are presented in Table 12. Following the clustering results determined by minimizing the Euclidean distance formula, the centroids' average for each cluster was subsequently calculated and is displayed in Table 13. In the conventional K-means analysis of the Whoscale Customer dataset, during the first testing iteration, the algorithm terminated at the 20th iteration. The K-means iteration stops when it has converged, meaning that the centroids of the clusters in the current iteration are the same as the centroids in the previous iteration.

### 5.2.2 WPM + K-means

The results of WPM + K-means testing on the Whoscale Customer dataset differ significantly from conventional K-means results. The iterations obtained by optimizing K-means using the WPM method are much fewer. In this study, 10 testing iterations were performed on the Whoscale Customer dataset. WPM values were used to initialize the initial centroids in K-means. In each test, three different WPM values were used to initialize the initial centroids: high, medium, and low. The results of the first iteration of K-means testing on the Whoscale Customer dataset are shown in Table 14. Table 15 shows the results of the 20th iteration of K-means testing on the Whoscale Customer dataset. The results of the WPM model calculations are displayed in Table 16. The test data, based on the recommendations of the WPM model, is presented in Table 17. Table 18 shows the results of the 7th iteration of WPM + K-means testing on the Whoscale Customer dataset.

**Table 12.** Initial centroids in the Whoscale Customer dataset for K-means

| Test Number | Test Data |
|---|---|
| 1 | 146, 221, 316 |
| 2 | 420, 425, 430 |
| 3 | 43, 57, 163 |
| 4 | 3, 184, 291 |
| 5 | 48, 56, 63 |
| 6 | 163, 202, 303 |
| 7 | 1, 11, 385 |
| 8 | 12, 51, 107 |
| 9 | 61, 62, 81 |
| 10 | 438, 439, 440 |

**Table 13.** The results of the K-means distance calculations on the Whoscale Customer dataset

| No | C1 | C2 | C3 | Cluster |
|---|---|---|---|---|
| 1 | 13379,42 | 11350,02 | 30641,14 | 2 |
| 2 | 8245,11 | 14738,00 | 30970,94 | 1 |
| 3 | 10091,15 | 15348,66 | 32873,53 | 1 |
| 4 | 20872,95 | 5759,47 | 35450,17 | 2 |
| 5 | 24053,11 | 11967,33 | 30051,92 | 2 |
| .. | …. | …. | …. | …. |
| .. | …. | …. | …. | …. |
| 20 | 14664,23 | 24663,71 | 27851,34 | 1 |
| 21 | 18029,41 | 7886,87 | 38089,51 | 2 |
| 22 | 21243,33 | 8746,82 | 33595,38 | 2 |
| 23 | 35988,60 | 24416,95 | 38205,26 | 2 |
| .. | …. | …. | …. | …. |
| .. | …. | …. | …. | …. |
| 907 | 18881,46 | 4925,84 | 37414,26 | 2 |
| 908 | 17051,53 | 12194,13 | 40291,26 | 2 |

**Table 14.** The results of the first iteration of K-means testing on the Whoscale Customer dataset

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **C1** | 1,709 | 2,535 | 3834,110 | 9320,787 | 13136,071 | 1516,016 | 5585,213 |
| **C2** | 1,126 | 2,544 | 14668,384 | 2944,551 | 3812,551 | 3591,588 | 852,704 |
| **C3** | 1,789 | 2,579 | 25299,684 | 26364,158 | 37336,368 | 5431,000 | 16202,105 |

**Table 15.** The results of the 20th iteration of K-means testing on the Whoscale Customer dataset

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **C1** | 1,306 | 2,537 | 7390,958 | 4439,769 | 6292,196 | 2495,534 | 2238,653 |
| **C2** | 1,160 | 2,573 | 32768,013 | 4827,680 | 5723,147 | 5535,920 | 1074,120 |
| **C3** | 1,964 | 2,536 | 11849,179 | 24717,107 | 33887,714 | 3409,321 | 15459,714 |

**Table 16.** WPM calculation result on the Whoscale Customer dataset

| No | Channel | Region | Fresh | Milk | Grocery | Frozen | Detergents_Paper | Delicassen | WP |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 12669 | 9656 | 7561 | 214 | 2674 | 1338 | 34117 |
| 2 | 2 | 3 | 7057 | 9810 | 9568 | 1762 | 3293 | 1776 | 33271 |
| 3 | 2 | 3 | 6353 | 8808 | 7684 | 2405 | 3516 | 7844 | 36615 |
| .. | … | …. | … | …. | … | … | …. | …. | …… |
| .. | … | …. | … | …. | … | … | …. | …. | …… |
| 194 | 2 | 3 | 180 | 3485 | 20292 | 959 | 5618 | 666 | 31205 |
| 195 | 1 | 3 | 7107 | 1012 | 2974 | 806 | 355 | 1142 | 13400 |
| 196 | 1 | 3 | 17023 | 5139 | 5230 | 7888 | 330 | 1755 | 37369 |
| 197 | 1 | 1 | 30624 | 7209 | 4897 | 18711 | 763 | 2876 | 65082 |
| 198 | 2 | 1 | 2427 | 7097 | 10391 | 1127 | 4314 | 1468 | 26827 |
| .. | … | …. | … | …. | … | … | …. | …. | …… |
| .. | … | …. | … | …. | … | … | …. | …. | …… |
| 333 | 1 | 2 | 22321 | 3216 | 1447 | 2208 | 178 | 2602 | 31975 |
| .. | … | …. | … | …. | … | … | …. | …. | …… |
| .. | … | …. | … | …. | … | … | …. | …. | …… |
| 439 | 1 | 3 | 10290 | 1981 | 2232 | 1038 | 168 | 2125 | 17838 |
| 440 | 1 | 3 | 2787 | 1698 | 2510 | 65 | 477 | 52 | 7593 |

**Table 17.** Initial Centroids in the Whoscale Customer dataset for WPM+K-means

| Test Number | Test Data |
|---|---|
| 1 | 96, 268, 493 |
| 2 | 468, 724, 772 |
| 3 | 83, 235, 847 |
| 4 | 285, 556, 747 |
| 5 | 189, 214, 766 |
| 6 | 131, 205, 311 |
| 7 | 6, 62, 73 |
| 8 | 23, 293, 561 |
| 9 | 96, 147, 883 |
| 10 | 716, 827, 893 |

**Table 18.** The results of the 7th iteration of WPM+K-means testing on the Whoscale Customer dataset

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **C1** | 1,253 | 2,546 | 8341,613 | 3779,893 | 5152,174 | 2577,238 | 1720,573 |
| **C2** | 1,136 | 2,593 | 36156,390 | 6123,644 | 6366,780 | 6811,119 | 1050,017 |
| **C3** | 1,962 | 2,472 | 7751,981 | 17910,509 | 27037,906 | 1970,943 | 12104,868 |

**Table 19.** The comparison of the number of iterations and DBI value on the Whoscale Customer dataset

| Test Data | K-means Iterations | WPM+K-means Iterations | K-means DBI Value | WPM+K-means DBI Value |
|---|---|---|---|---|
| 1 | 20 | 7 | 0,678 | 0,514 |
| 2 | 22 | 9 | 0,667 | 0,538 |
| 3 | 20 | 8 | 0,917 | 1,637 |
| 4 | 32 | 10 | 0,610 | 0,599 |
| 5 | 16 | 7 | 0,750 | 0,618 |
| 6 | 21 | 10 | 0,910 | 0,682 |
| 7 | 18 | 8 | 0,703 | 0,751 |
| 8 | 27 | 9 | 0,566 | 0,587 |
| 9 | 35 | 10 | 1,775 | 0,771 |
| 10 | 22 | 7 | 0,942 | 0,571 |
| Average | 23,3 | 8,5 | 0,852 | 0,727 |

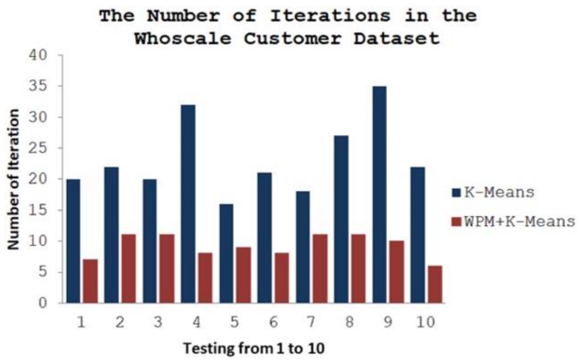

**Figure 6.** The comparison of the number of iterations

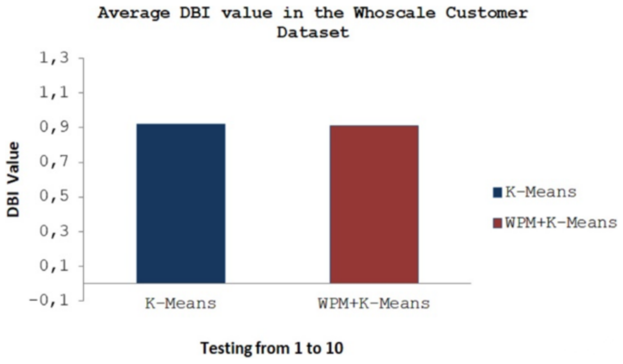

**Figure 7.** The comparison of the number of iterations

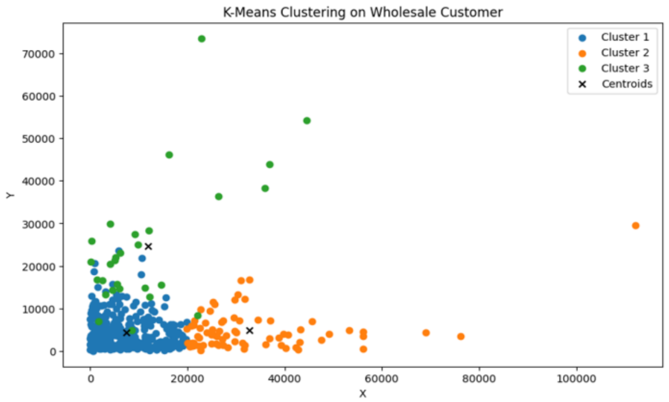

**Figure 8.** The clustering results on K-means on Whoscale Customer dataset

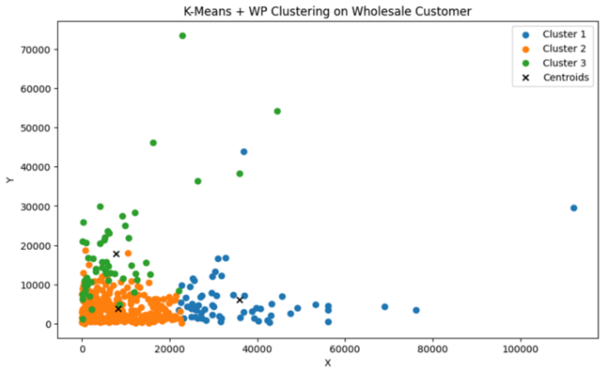

**Figure 9.** The clustering results on WPM + K-means on Whoscale Customer dataset

**Table 20.** The comparison of the number of iterations and DBI Value on the Captured Fisheries dataset

| Test Data | K-means Iterations | WPM + K-means Iterations | K-means DBI Value | WPM+K-means DBI Value |
|---|---|---|---|---|
| 1 | 11 | 5 | 1,882 | 0,842 |
| 2 | 11 | 5 | 0,645 | 0,825 |
| 3 | 11 | 3 | 1,954 | 0,927 |
| 4 | 10 | 5 | 1,867 | 0,885 |
| 5 | 11 | 7 | 0,708 | 0,982 |
| 6 | 10 | 4 | 0,935 | 0,895 |
| 7 | 13 | 5 | 1,458 | 0,889 |
| 8 | 9 | 4 | 0,826 | 0,915 |
| 9 | 8 | 4 | 0,743 | 0,999 |
| 10 | 4 | 4 | 1,206 | 0,946 |
| Average | 9,8 | 4,6 | 1,222 | 0,910 |

5.2.3 Comparison of DBI values

In the conducted experiments on the dataset, a comparison was made between the traditional K-means and the WPM + K-means methods. The number of iterations required for convergence varied significantly between the two approaches. For K-means, the number of iterations ranged from 16 to 35 across the ten tests, with an average of approximately 23.3 iterations. In contrast, when applying the WPM + K-means method, the number of iterations was consistently lower, ranging from 7 to 10 iterations across the tests and averaging around 8.5 iterations.

Assessing the clustering quality using DBI, it became evident that the WPM + K-means approach consistently outperformed conventional K-means. The DBI values for WPM + K-means were consistently lower, indicating better cluster separation and cohesion. On average, the DBI value for K-means was 0.852, while for WPM + K-means, it was 0.727, suggesting that the latter method produced more coherent and well-separated clusters in the dataset. These results underscore the effectiveness of the WPM + K-means approach in optimizing clustering performance on the Whoscale Customer dataset, as shown in Table 19 and Figures 6-9.

### 5.3 Captured Fisheries dataset

In the conducted experiments on the Captured Fisheries dataset, we conducted a comparative analysis between the traditional K-means method and the WPM + K-means approach. One of the notable distinctions between these methods was the number of iterations required for convergence. For K-means, the number of iterations displayed considerable variability across the ten tests, spanning from 16 to 35 iterations.

On average, K-means converged in approximately 23.3 iterations. In contrast, the utilization of the WPM + K-means approach consistently resulted in a lower number of iterations. The iterations consistently fell within the narrow range of 7 to 10 iterations across all tests, with an average of approximately 8.5 iterations. This efficiency in convergence implies that the WPM + K-means method significantly reduces computational effort and time compared to the conventional K-means approach. To assess the quality of clustering achieved by these methods, we employed DBI. Remarkably, the WPM + K-means approach consistently outperformed the conventional K-means method when considering the DBI values. On average, K-means yielded a DBI value of 0.852, indicating relatively weaker cluster quality. In contrast, the WPM + K-means method exhibited significantly superior results, with an average DBI value of 0.727. These findings underscore the effectiveness of the WPM + K-means approach in producing more cohesive and well-defined clusters within the Captured Fisheries Dataset, highlighting its potential for enhancing clustering performance in similar contexts, as shown in Table 20, Figures 10-13.

A lower DBI signifies superior clustering performance since it indicates an equilibrium between cluster separation and cohesion. Well-separated clusters that maintain internal cohesion are preferred as they aptly capture the intrinsic data structure, fostering insightful interpretation and analysis.

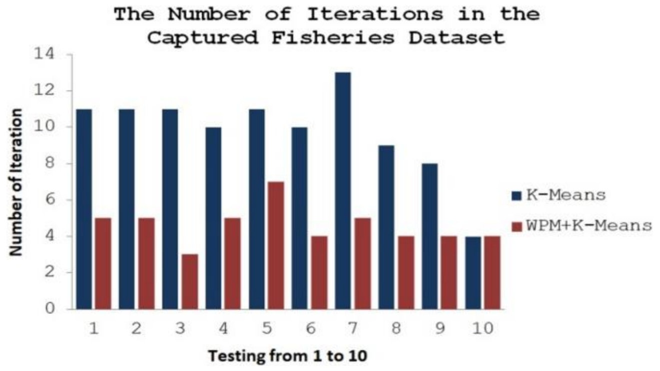

**Figure 10.** The comparison of the number of iterations

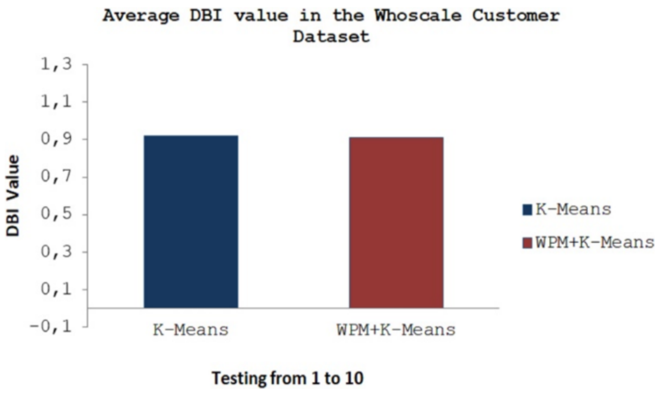

**Figure 11.** The comparison of the number of iterations

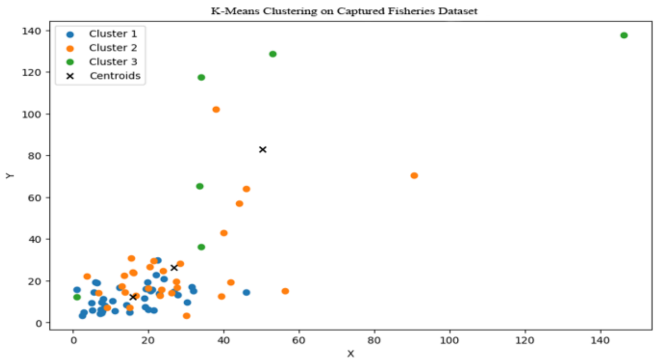

**Figure 12.** The clustering results on WPM+K-means on Captured Fisheries dataset

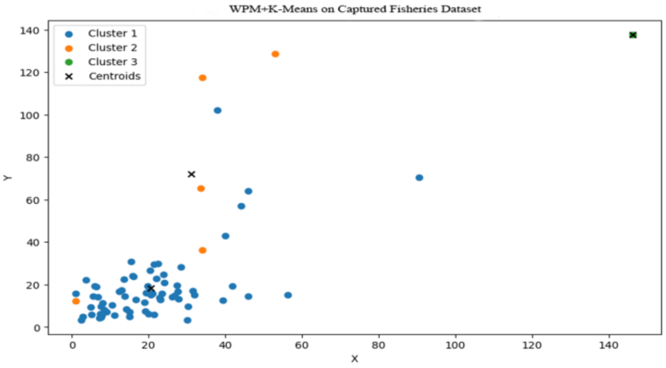

**Figure 13.** The clustering results on WPM+K-means on Captured Fisheries dataset

## 6. CONCLUSION

This study effectively demonstrates the optimization of K-means clustering performance using WPM. By comparing K-means performance with WPM + K-means, the research reveals notable improvements in cluster evaluation across three distinct datasets. For the QSAR Dataset, the average DBI value improved from 0.852 with conventional K-means to 0.727 with WPM + K-means, accompanied by a significant reduction in the average number of iterations from 23 to 8. Similarly, for the Whoscale Customer dataset, the average DBI value slightly improved from 0.921 to 0.910, with a decrease in the average number of iterations from 23 to 10. Notably, the captured fisheries dataset showed significant enhancement, with the average DBI improving from 1.222 to 1.052 and the average number of iterations decreasing from 9 to 4. These results underscore the effectiveness of the Weight Product Model in optimizing cluster evaluation values and reducing the computational burden of the K-means algorithm. Lower DBI values indicate better clustering performance, highlighting the potential of WPM + K-means in improving the quality and efficiency of K-means clustering across diverse datasets. Importantly, this study provides valuable insights into enhancing K-means clustering performance through the integration of the Weighted Product method, offering potential applications in various domains and data analysis tasks.

## ACKNOWLEDGEMENT

The authors extend their appreciation to Malikussaleh University for funding this research.

## REFERENCES

[1] Jahwar, A.F., Abdulazeez, A.M. (2020). Meta-heuristic algorithms for K-means clustering: A review. PalArch's Journal of Archaeology of Egypt/Egyptology, 17(7): 12002-12020.

[2] Ding, N., Xu, Y., Tang, Y., Xu, C., Wang, Y., Tao, D. (2022). Source-free domain adaptation via distribution estimation. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, pp. 7202-7212. https://doi.org/10.1109/CVPR52688.2022.00707

[3] Zhao, D., Hu, X., Xiong, S., Tian, J., Xiang, J., Zhou, J., Li, H. (2021). K-means clustering and kNN classification

based on negative databases. Applied Soft Computing, 110: 107732. https://doi.org/10.1016/j.asoc.2021.107732

[4] Hossain, M.Z., Akhtar, M.N., Ahmad, R.B., Rahman, M. (2019). A dynamic K-means clustering for data mining. Indonesian Journal of Electrical Engineering and Computer Science, 13(2): 521-526. https://doi.org/10.11591/ijeecs.v13.i2.pp521-526

[5] Borlea, I.D., Precup, R.E., Borlea, A.B. (2022). Improvement of K-means cluster quality by post processing resulted clusters. Procedia Computer Science, 199: 63-70. https://doi.org/10.1016/j.procs.2022.01.009

[6] Ros, F., Riad, R., Guillaume, S. (2023). PDBI: A partitioning Davies-Bouldin index for clustering evaluation. Neurocomputing, 528: 178-199. https://doi.org/10.1016/j.neucom.2023.01.043

[7] Wijaya, Y.A., Kurniady, D.A., Setyanto, E., Tarihoran, W. S., Rusmana, D., Rahim, R. (2021). Davies bouldin index algorithm for optimizing clustering case studies mapping school facilities. TEM Journal, 10(3): 1099-1103. https://doi.org/10.18421/TEM103-13

[8] Sinaga, K.P., Yang, M.S. (2020). Unsupervised K-means clustering algorithm. IEEE Access, 8: 80716-80727. https://doi.org/10.1109/ACCESS.2020.2988796

[9] Ikotun, A.M., Ezugwu, A.E., Abualigah, L., Abuhaija, B., Heming, J. (2022). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences, 622: 178-210. https://doi.org/10.1016/j.ins.2022.11.139

[10] Ahmed, M., Seraj, R., Islam, S.M.S. (2020). The K-means algorithm: A comprehensive survey and performance evaluation. Electronics, 9(8): 1295. https://doi.org/10.3390/electronics9081295

[11] Govender, P., Sivakumar, V. (2020). Application of K-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). Atmospheric pollution research, 11(1): 40-56. https://doi.org/10.1016/j.apr.2019.09.009.

[12] Uddin, M.A., Roy, S. (2023). Examining TOD node typology using k-means, hierarchical, and latent class cluster analysis for a developing country. Innovative Infrastructure Solutions, 8(11), 304.

[13] Zhuang, Y., Chen, X., Yang, Y. (2022). Wasserstein K-means for clustering probability distributions. Advances in Neural Information Processing Systems, 35: 11382-11395.

[14] Rezaee, M.J., Eshkevari, M., Saberi, M., Hussain, O. (2021). GBK-means clustering algorithm: An improvement to the K-means algorithm based on the bargaining game. Knowledge-Based Systems, 213: 106672. https://doi.org/10.1016/j.knosys.2020.106672

[15] Nguyen, T.H.T., Dinh, D.T., Sriboonchitta, S., Huynh, V.N. (2019). A method for K-means-like clustering of categorical data. Journal of Ambient Intelligence and Humanized Computing, 14(11): 15011-15021. https://doi.org/10.1007/s12652-019-01445-5

[16] Aldino, A.A., Darwis, D., Prastowo, A.T., Sujana, C. (2021). Implementation of K-means algorithm for clustering corn planting feasibility area in south lampung regency. Journal of Physics: Conference Series, 1751(1): 012038. https://doi.org/10.1088/1742-6596/1751/1/012038

[17] Barile, C., Casavola, C., Pappalettera, G., Kannan, V.P. (2022). Laplacian score and K-means data clustering for damage characterization of adhesively bonded CFRP composites by means of acoustic emission technique. Applied Acoustics, 185: 108425. https://doi.org/10.1016/j.apacoust.2021.108425

[18] Dinata, R. K., Hasdyna, N., Retno, S., Nurfahmi, M. (2021). K-means algorithm for clustering system of plant seeds specialization areas in east Aceh. ILKOM Jurnal Ilmiah, 13(3): 235-243. https://doi.org/10.33096/ilkom.v13i3.863.235-243

[19] Rengasamy, S., Murugesan, P. (2022). K-means–Laplacian clustering revisited. Engineering Applications of Artificial Intelligence, 107: 104535. https://doi.org/10.1016/j.engappai.2021.104535

[20] Boucetta, C., Hussenet, L., Herbin, M. (2023). Improved Euclidean distance in the K nearest neighbors method. In International Conference on Innovations for Community Services, pp. 315-324. https://doi.org/10.1007/978-3-031-40852-6_17

[21] Hatefi, M.A. (2023). A typology scheme for the criteria weighting methods in MADM. International Journal of Information Technology & Decision Making, 22(04), 1439-1488. https://doi.org/10.1142/S0219622022500985

[22] Stallmann, M., Wilbik, A., Weiss, G. (2024). Towards unsupervised sudden data drift detection in federated learning with fuzzy clustering. In 2024 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Yokohama, Japan, pp. 1-8. https://doi.org/10.1109/FUZZ-IEEE60900.2024.10611883