HYPERPARAMETERS AND CENTROID IMPROVEMENTS IN THE K-MEDOIDS METHOD FOR GROUPING PROCESSED BEEF SMEs

Article · February 2024

DOI: 10.28919/cmbn/8369

1 author:

Fitri Agustina
Institut Teknologi Sepuluh Nopember
30 PUBLICATIONS **55** CITATIONS

SEE PROFILE

Available online at http://scik.org

Commun. Math. Biol. Neurosci. 2024, 2024:13

https://doi.org/10.28919/cmbn/8369

ISSN: 2052-2541

HYPERPARAMETERS AND CENTROID IMPROVEMENTS IN THE K-MEDOIDS METHOD FOR GROUPING PROCESSED BEEF SMEs

BAIN KHUSNUL KHOTIMAH[1,*], FITRI AGUSTINA[2], OKTAVIA RAHAYU PUSPITARINI[3], ANDHARINI DWI CAHYANI[1], YENI KUSTIYAHNINGSIH[1], DEVIE ROSA ANAMISA[1]

[1]Department of Informatics Engineering, Trunojoyo Madura University, Bangkalan, 69162, Indonesia

[2]Department of Industrial Engineering, Trunojoyo Madura University, Bangkalan, 69162, Indonesia

[3]Departmen of Animal Husbandry, Islamic University of Malang, 65144, Indonesia

Abstract: Hyperparameter tuning is a crucial step in finding the optimal machine learning parameters by iterative processing depending on the proper tuning, resulting in maximum accuracy. K-Medoids is a clustering algorithm susceptible to centroids and determining the correct K value to produce the low error. The most straightforward hyper-parameter technique used is Grid Search (GS) because it searches for deals using a specific range of distances. The K-Medoids method is used to form low-bias cluster models and make decisions on cluster development strategies that are appropriate to conditions in SMEs. Data from survey results is processed through the Label Encoding stages to convert categorical data into numerical data, Data Imputation to fill in empty values in the data using the mean method, and Data Normalization using the Min-Max method to standardize data in the range 0 to 1 for optimal processing. The results of cluster optimization using the GS method using a specific value range value is at the optimal number of clusters k = 10, with the lowest SSE value of 32,0970. When an increase in optimization for centroids in K-Medoids

*Corresponding author

E-mail address: bain@trunojoyo.ac.id

Received November 29, 2023

using a binary search algorithm results in an optimal cluster at k = 3 of DBI 0.1021 value. A comparison of the performance of the K-Means method and K-Medoids method shows that k-medoids produce the lowest SSE in the best parameter optimization through GS to improve model efficiency.

Keywords: clustering; grid search; K-medoids; binary search; sum of squared errors (SSE); SMEs

2024 AMS Subject Classification: 92B05.

1. INTRODUCTION

Fundamental changes in the processed beef production and trade system have implications for regions and industrial clusters [1]. The livestock sector in the Madura region, mainly processed food products, is an essential factor in supporting the community's economy. The solution offered to solve the problem of determining the quality of processed food products, especially yeast jerky, is to group them into produce groups that will be validated by nutrition and food experts [2]. Determining the quality of processed beef depends on the attributes used in mapping the quality of processed food from Madurese beef yeast meat, namely body condition consisting of age (years), meat quality, type of cow, feeding pattern, and features of meat processing techniques based on drying time, and duration of storage [3].

Non-hierarchy-based clustering varies Mechine Learning (ML) greatly depending on the centroid update process [4]. Clustering algorithms for applications ML of various types of data, both numeric and categorical [5][6], with partition grouping being further divided into center-based algorithms [7] with means [8], different harmonic methods [9], medoids algorithm [10], and spectral clustering algorithm [11]. K-Means is a non-hierarchical technique for grouping data that looks for to separate information into multiple groups, or clusters, with the goal of grouping data that share similar attributes into a single cluster. Data with distinct features, on the other hand, are categorized into different groups [12]. Medoid objects are used as cluster centers in the k-medoids method, a partitioning technique used in cluster analysis [13]. The medoids object is chosen two as the middle point of the data group in the cluster [14]. The data that has been obtained will be analyzed using the k-medoids clustering algorithm to group it based on the required guidance. The k-medoids method uses actual objects from the data set as cluster centers, while the k-means

method uses the average for resulting cluster representation [15]. In addition, the k-means method often faces convergence problems when the data is extensive. To determine the most accurate way, we will use the Sum of Squared Errors method as the most accurate method. The sum of Squared Errors (SSE) is a statistical method used to measure the total difference between the actual and obtained values. This method is useful in analyzing the accuracy or error of predictions in a model or algorithm. The squaring difference between the real value and the predicted value, SSE shows how much the model or algorithm accurately describes or indicates the data. The smaller the SSE value, the better the model or algorithm matches the data to the actual value [16][17].

In this research, K-medoids were chosen because this algorithm effectively clusters data that contains outliers, considering that the existing data has values significantly different from the average of other data. [18]. K-Medoids is a variant of k-means that is sensitive to outliers. In the k-means method, objects with extreme values can significantly influence the data distribution. However, K-Medoids have reduced sensitivity to outliers by not relying on centroids to represent cluster centers. Apart from that, the silhouette coefficient is used to determine the optimal number of k. The Silhouette coefficient value ranges from -1 to 1 and indicates the extent to which data grouped in a cluster is similar. If the average value of the Silhouette coefficient is close to 1, it can be considered that the cluster has good quality. On the other hand, if the average value of the Silhouette coefficient is relative to -1, it can be concluded that the cluster has poor quality [19].

The number of clusters in a data set must be determined a priori, and the initial cluster center *m* (medoids) must be selected randomly. They affect algorithm performance, mainly if applied to large data sets. Determining the optimal number of clusters to start clustering is complicated because random selection of initial cluster centers (centroids) sometimes results in minimal local convergence [20]. Grid search (GS) hyperparameters are needed for selecting *k* and the best number of medoids based on iteration stops. The usage for determining the cluster (centroid) contains different initial binary searches to get optimal cluster results [21][22].

The object matches its cluster poorly and neighboring clusters poorly when the value is high. A grouping configuration is appropriate if the majority of the objects have elevated values. The clustering configuration may have too many clusters if a large number of the points have low or

negative values [23]. The Silhouette coefficient considers the proximity of the data to the cluster in which it resides and the separation of the data from other clusters as a more holistic measure of clustering quality. The K-Medoids algorithm collects data based on similarity so that data with the same characteristics will be put into the same cluster. The similarity of the data can be measured by how close the distance between the data is and the distance between the data and the data centroid [24]. The working principle of the K-Medoids algorithm is to determine the number of clusters first and then determine the centroid point of the data randomly [25]. After that, allocate the data to the nearest clusters, and the process will be repeated until it finds a stable centroid. The output of the K-Medoids algorithm is highly dependent on determining the number of clusters and centroid selection, which is selected randomly and repeatedly. The problem with the K-Medoids algorithm is that it produces a final centroid that is not the true cluster center. This algorithm must be run many times with different initial centroids to get the final centroid that is considered the best [26][27][28].

The dynamic cluster algorithm provides better and more accurate potential segmentation results within the K-means algorithm and calculates the number of clusters (k) to produce optimal cluster quality [29]. Though it still requires a random selection for the centroid point in the clustering process, this algorithm shares some drawbacks with the K-means algorithm. Identifying the centroid point is finished by transforming K-means into K-means Binary Search Centroid (KBSC), which employs the Binary Search technique approach to determine the centroid point [30][32]. The study's findings demonstrate that the K-means Binary Search Centroid (KBSC) algorithm outperforms the K-means algorithm regarding intra- and inter-cluster values. Nevertheless, there are restrictions on how many clusters the K-means Binary Search Centroid (KBSC) algorithm can form. According to an explanation, there are advantages in identifying the starting cluster center and drawbacks in determining the number of clusters for both the Dynamic K-means and KBSC algorithms. Other than that, the Dynamic K-means algorithm has advantages when figuring out the number of clusters and disadvantages when figuring out cluster centers [32][33][34].

The research proposed that determining the hyperparameter k value offered to combine the K-Medoids Algorithm with the Binary Search Centroid (KMBSC) can complete the clustering

process to determine the number of clusters to be formed. The clustering process in terms of determining the number of clusters, determining the number of clusters, and determining cluster centroid point to produce the SSE value of the index value for the Beef Processing MSME clustering case study. The k-Medoids algorithm with the Binary Search Centroid (KMBSC) algorithm has limitations in determining the number of clusters to be formed based on the centroid having better intra- and inter-cluster values than random centroid update. Therefore, it is proposed to combine the K-medoids algorithm by optimizing the number of m pada k-medoids and clusters to complement the clustering process in assuming the number of clusters and determining points cluster centroid. So, using measurement with the Davies-Bouldin Index value is the best case study for clustering the quality of processed meat in SMEs [35][36].

2. RESEARCH METHODOLOGY

2.1 Dataset Description

Data is taken from government agencies and the beef processing SME community, consisting of annual turnover, condition of cattle, meat quality, price, etc. The sustainability of the production process is more guaranteed. The asset value SME are also described using nominal rupiah, then simplified using numbers and grouping using k-medoids.

Table 1. Criteria for beef processing in SMEs

Criteria	Encoding label	Value min max
Type of Cattle	Categorical	1-6
Omset	Rp (Million)	5-100
Age of Cattle	Year	2-6
Weight	kg	100-1000
Price of Meat	Rp (Thousand)	120-150
Length of Cattle	cm	120-200
Vaccine	Yes/No	1/2
Type of feed	Categorical	1-4
Types of yeast	Synthetic/natural	1/2
Long drying time	Hour	1-30
Storage time	Day	120
Types of yeast	Synthetic/natural	1/2

The next stage is processing operational data to extract information, converting categorical data to numeric, and changing the data scale to a specific range of values using the scale encoding.

2.2 Preprocessing Data

Missing Completely at Random (MCAR) refers to the random occurrence of missing data, where the distribution of missing data on a feature is independent of the observed or missing data. This method generates missing data randomly based on a predetermined proportion while using the entire dataset. This approach has the benefit of facilitating researchers' estimation of the computational performance of the suggested model. Another mechanism is Missing at Random (MAR), in which the observed data is independent of the missing data. Still, the distribution of missing data on a feature depends on the observed data. Finally, Not Missing at Random (NMAR), in which the missing data determines how the missing data on a feature is distributed [28]. The most typical common and simple method to replace missing is mean imputation [29].

$$Ximp = \frac{x_1 + x_2 + x_3 + \cdots + x_i}{N} \quad (1)$$

With X_{imp} is observations $x_1, x_2, \ldots, x_i$ from the dataset without missing values, N is the total number of observations that do not include missing values. This technique is straightforward and effective when data is Missing Completely at Random (MCAR) [30].

Normalization is a process to change values so that all values in the data have uniform values with the same range. This normalized data will later become input to the clustering process [5]. The min-max normalization process can be calculated as follow:

$$X_{norm} = \frac{X' - \min(x)}{\max(x) - \min(x)} \left(new_{max}(x) - new_{min}(x)\right) + new_{min}(x) \quad (2)$$

The data normalization process is obtained using the min-max normalization method, for each value on an attribute is reduced by the minimum value on that attribute, then divided by the range value.

2.3 K-Medoids Clustering

In general, there are two clustering approaches methods, namely, the partition approach and the hierarchical approach. Clustering with a partition approach is a grouping of data from one large group and then divided into several smaller groups [8]. An example of a clustering method with a

partition approach is K-Means Clustering. Clustering with a hierarchical system, often called Hierarchical Clustering groups data by combining each record or individual in the data into clusters. An example of a clustering method with a hierarchical approach is Agglomerative Hierarchical Clustering [24].

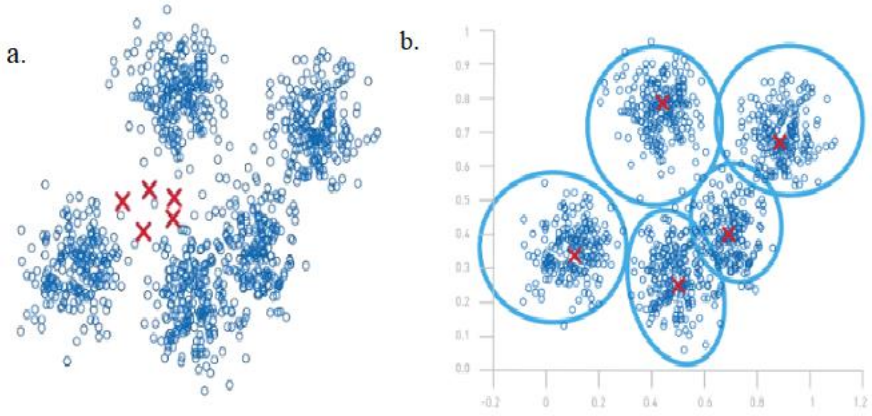

Figure 1. a. Determining centroids data; b. clustered data [11].

The k-medoids method divides data consisting of n objects into k clusters, where the number of k is not greater than n. Medoids are used as cluster representations and function as cluster centers. The process of forming clusters in the k-medoids algorithm is carried out by calculating the similarity distance between medoids and non-medoid objects. This analysis aims to minimize the dissimilarity of each object in the cluster by using the absolute error value (E) [17].

$$E = \sum_{c=1}^{k} \sum_{i=1}^{n_c} |p_{ic} - O_c| \quad (3)$$

with n_c = The number of objects in the c-th cluster, p_{ic} = Non-medoid object i in the c-th cluster and O_c = Value of medoids in the c-th cluster. So, randomly select an object as a point that is not a medoid. Next, calculate the distance of the object in each cluster to the non-medoid candidates, to produce variants. The smallest variance value (S), if the new TD <old TD, swap the position of the new medoid, then it becomes a new medoid. So that the final results of the medoid do not change [27][28].

Algorithm k-medoids

Input:

Number of variations of cluster c_n

A dataset MSE n feature

Output: the best cluster (c)

the best stop iteration

The minimized dissimilarities of each object

Algorithm:

1. Initialize the similarity calculation process for all data in setting *c* according to n random data variant points from space D.
2. Determine each calculated data point based on its closest medoid (m).
3. Calculate the medoid on each data variant for the next iteration
4. Updating iteration:
    a. Randomly select another non-medoid object for comparison with the next iteration
    b. Swap the medoid (m) with the data point (or) calculate the total cost (tc) every total data.
5. Select the medoid with the lowest cost in the form of the lowest error measurement.

Termination:

If it matches the best model, then the k value is concluded, and if you still need to find a model, go to step 1. So, the medoid obtained the model with the lowest error.

2.4 Grid Search

Grid Search (GS) is a simple and capable search method in a high-dimensional hyperparameter configuration space, as the number of judgments increases exponentially as the hyperparameter search frequency increases. Hyperparameters are obtained by assuming that k parameters exist and each has n separate values. The computational complexity increases exponentially at a rate of O(nk) [20][21]. Thus, GS can be an efficient HPO approach as a thorough exploration or brute force method that tests all combinations of hyperparameters given a grid configuration.

GS operates by assessing the cartesian product of a finite set of values the user specifies [22]. GS alone will not further exploit areas that perform well. Therefore, the following process must be performed manually to identify the global optimum point. The GS workflow is presented in the steps follows:

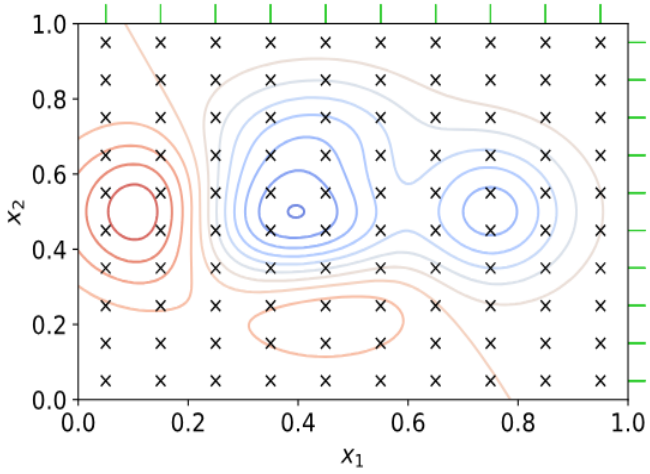

Figure 2. GS Process Ilustration [22].

1. **Initialization.** Determine and evaluate some serval points of the configuration space for each parameter.
2. **Adjusting data.** Using the value of each point based on a probabilistic distribution evaluates the performance of the most optimal method.
3. **Earn points on each possible next point**. Obtain the next promising issue x_{i+1} through optimization of the acquisition function on the Algorithm performance.
4. **Evaluate**. Evaluate the selected point by calculating the objective function to get y_{i+1}.
5. **Update.** Update data D at n+1.
6. **Repeat** step 2; step 5 until sampling is required according to the number of points n, next until the lowest cost and repeat again.

2.5 Binary Search Algorithm

Binary search is a method of searching for data in an ordered array. This method is more efficient than the linear search method, where all elements in the array are tested one by one until the desired part is found. Apart from binary search, there are also interpolation search and jump search, which

both work on sorted data. The searching on sorted data results in a fast search, with interpolation search having an average time complexity of O(log log n), while jump search is O(kn1). jump search is O(kn1/(k+1)). The time complexity for binary search is O(log n), as proposed by Knuth. [29]. The division process will continue until the data is found [30]. The principle of binary search can be explained as follows: Suppose the left index is i and the right index is j.

1. Initially i with one and j with n. with n. Divide the two array elements on the center element. The center element is the element with index k = (i + j) div 2. (The center element, L[K], divides the array into two parts, namely the left part L[i..j] and the right part L[k+1..j])
2. Check if L[K] = x, if L[K] = x then the search is complete because x has been found. But if L[K] ≠ x, it must be determined whether the search will be done in the left array or the suitable array.
3. Done in the left array or the exemplary array. If L[K] < x, the search is done again on the left array.
4. Conversely, if L[K] > x, then the investigation is conducted again on the suitable array.
5. Repeat the first step until x is found or i > j, i.e., the array size is zero.

2.6 Proposed Algorithm

K-medoid clustering is a method in unsupervised learning, the same as k-means clustering. So, the clustering process is set to find cluster center points (centroids) that minimize the distance between members in the cluster and the center point. In a population set *x*, several data {x1, x2, x3, …, xn}. Furthermore, the data will be grouped into clusters with the number of clusters being *c*, in this case $c \leq n$. In K-medoid clustering, set members are grouped based on their proximity to each other so that the average distance of members in the cluster is minimal. In K-medoid clustering, the medoid concept is known. Medoid is a cluster member, which is the central point of the cluster. The number of medoids in the population is equal to k. Thus, the set M can be symbolized as $\{m_1, m_2, m_3, \ldots, m_n\}$. This algorithm aims to minimize the number of similarities between each object and its corresponding reference point.

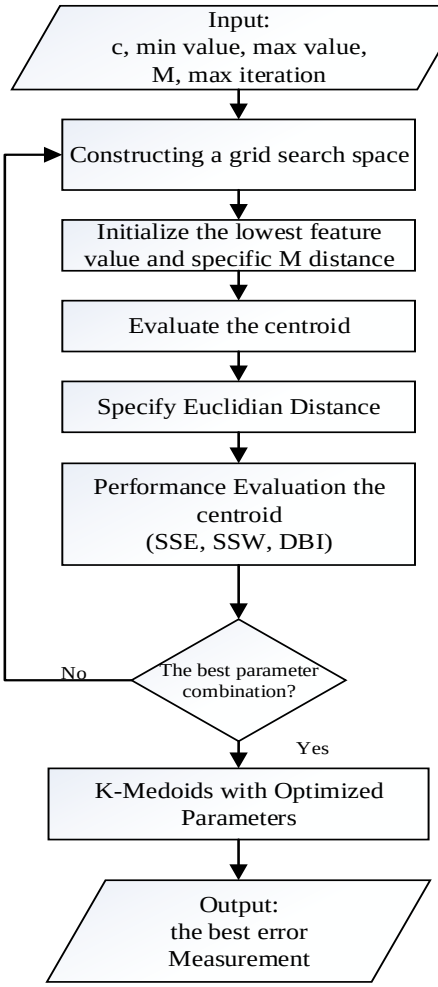

Figure 3. The proposed methodology's framework in graphical representation.

We have combined several algorithms, namely the K-medoids algorithm with the binary search algorithm, to improve centroid updates and hyperparameter grid search for parameter search, which aims to eliminate the computational burden. The proposed hybrid algorithm is explained as follows:

1. **Initialize parameters**: c (cluster), Initialize max value.
2. **Contructing a grid search**: determine the number *c* of clusters =[10,20,30,40] and Centroid range =[0.01, 0.1, 1.0].
3. **Initialize the lowest feature value and specific M distance**: binary search stage with the first calculation of the distance between centroid points.

$$M = \frac{\max(a_i) - \min(a_i)}{c} \quad (4)$$

4. **Evaluate the centroid:** sorting the value of each feature, from largest to smallest in each data.Evaluating closest all data/objects closest to the centroid most relative to the data/object.

$$C_k = \min(a_i) + (k-1)M \quad (5)$$

5. **Specify Euclidian Distance**: the distance to the center point with smallest s (centroid) on each data record.

$$D = \sqrt{(x_{1i} - x_{1j})^2 + (x_{ni} - x_{cj})^2} \quad (6)$$

6. **Updating Solutions:** Update the centroid every *n* iterations in order of centroids from smallest to next.
7. **Acceptability of the solution:** determining the smallest centroid distance. Otherwise, maintain the current solution.
8. **Termination of Algorithm**: Algorithm Termination: Stops the algorithm based on specified criteria if the performance measurements have been met, resulting in the best solution. Otherwise, then go back to step 2.

2.7 Evaluation Measures

The most accurate and appropriate algorithm for assessing algorithm performance is found through experimental scenarios in the model's evaluation. Following the formation of the cluster results, the algorithms are compared, and conclusions are made regarding which algorithm performs best, which has the best algorithm error, and what the ideal number of clusters is based on test data criteria. The sum of Square Within a Cluster (SSW) is a formula used to measure cohesion within an i cluster. The procedure is stated as follows [13].

$$SSW_i = \frac{1}{m_i} \sum_{j=i}^{m_i} d(x_i, c_i) \quad (7)$$

The sum of Square Between Clusters (SSB) is a formula used to measure separation between clusters, the procedure is as follows:

$$SSB_{ij} = d(c_i, c_j) \quad (8)$$

After getting the cohesion and separation values, the ratio (Rij) is measured to compare the ith cluster with the jth cluster. A good cluster is a cluster that has the smallest possible cohesion value and the most significant possible separation value. The formula for calculating the ratio (Rij) is as follows:

$$R_{ij} = \frac{SSW_i + SSW_j}{SSB_{ij}} \quad (9)$$

We can calculate the Davies-Bouldin Index (DBI) [9] with this ratio value using the following formula:

$$DBI = \frac{1}{c} \sum_{i=1}^{k} max_{i \neq j} (R_{ij}) \quad (10)$$

The k value is the number of clusters used in the analysis. The smaller the DBI value obtained (non-negative and >= 0), the better the quality of the clusters resulting from the K-Medoids grouping used [12]. Measuring errors in each cluster using SSE (Sum of Squared Errors) is a measure used to measure how far data points in a cluster are from the cluster center or centroid in clustering analysis. SSE calculates the sum of the squares of the distance between each data point and its cluster center. Deviation measures the extent to which each data in a group or cluster differs from the cluster center, and the deviation metric is used [13].

$$SSE = \sum_{k-1}^{k} \sum xi \in sk \, ||x_i - c_k||_2^2 \quad (11)$$

With x_i = Value of the – ith data feature, c_k = Feature or attribute of the ith cluster center point. The smaller the SSE value, the better the clustering quality because it shows that the data points in the cluster are closer to the cluster center. Therefore, SSE is used as one of the criteria for evaluating the grouping quality in clustering.

3. MAIN RESULTS

We implemented the algorithm with the Python language jupyter notebook program. We ran it on an Intel core i7 machine with g10700K 3.8Ghz Up To 5.1Ghz 16MB Cache. The measurement comparison results of convergence time, size, and fitness values are obtained from GS-K-Means [21], GS-K-Medoids [22], GS-KMBS, proposed method GS-KMBSC are shown in Tables 2, and Table 3.

Table 2. Results obtained for DBI measure

C	GS-K-Means	GS-K-Medoids	GS-KBSC	GS-KMBSC
2	0.5912	0.8713	0.6912	0.2912
3	0.0921	0.9425	0.0921	0.1021
4	0.7910	0.6312	0.0903	0.3120
5	0.6791	0.5728	0.8032	0.9307
6	0.2501	0.6602	0.5021	0.1911
7	0.2610	0.5324	0.8483	0.3923
9	0.3701	0.3211	0.1522	0.6122
10	0.9872	0.4025	0.3321	0.8304

The research results show that the k-medoids hyperparameter algorithm with Binary Search Centroid has the advantage of constant iteration in each test, with a DBI value close to 1. This is different from the test results of the traditional k-means algorithm where the DBI value is higher because it depends on the initialization of the center point value The initial clusters are random so that they obtain different validation values. Table 3. shows the optimal cluster results with a DBI value = 0.1021 during the 3-th cluster, with the best epoch each learning. Grid Search experiments to find parameters that are close to optimal in combinations within a given range. The process has been time-consuming if the dimensional data set is relatively high or the number of parameter combinations is enormous. Therefore, even so GS provides excellent results in almost any data set but is only reliable in low-dimensional data sets with few parameters. Each algorithm is implemented and executed in 5 different runs, each with a specific max iteration limit using the parameters. Table 3. shows a comparison of the computing time of several methods. The results show that the fastest computing time used several methods. Hyperparameter method in k-medoid looks for the best medoids. So, traditional k-means use parameters with specific iterations to obtain the lowest SSE.

Table 3. Grid Search Experiment results on several clustering methods

No	Methods	Iteration	c (cluster)
1	**GS-K-Means**	398	4
2	**GS-K-Medoids**	874	6
3	**GS- KBSC**	321	4
4	**GS-KMBSC**	534	8

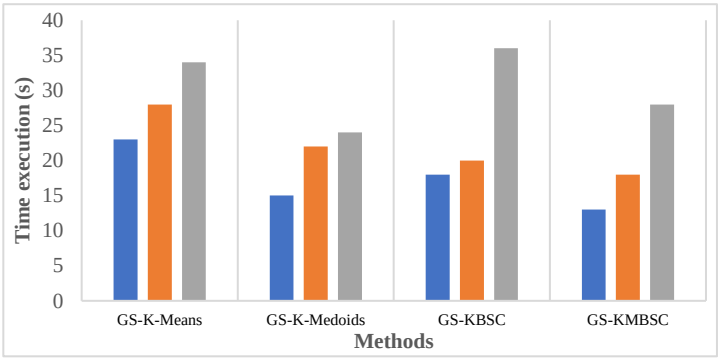

Figure 4. The measurement of computing time (s) testing multiple methods.

Figure 4. K-Means takes an average of 56 seconds for 3 clusters, while data processing in K-Medoids takes 1 minute 38 seconds. So, the higher the iteration and grouping specified dependence, the longer the data processing. Adding the centroid refinement process with binary search reduces iteration because the lowest centroid, stored in the local search, is selected without continuously repeating random values.

4. DISCUSSION

The results of research testing to determine the performance of fundamental differences between k-means and k-medoids using SSE. Hyperparameter pada K-Medoids algorithm yields better results in average cluster quality compared to traditional K-Mean in SSE. The development of the SSE value is in line with the increase in the number of clusters used in the experiment. The range of clusters explored starts from 2 to 10, based on SSE show graph that the lowest SSE, that found when using 10 clusters, with an SSE value of 32.0970.

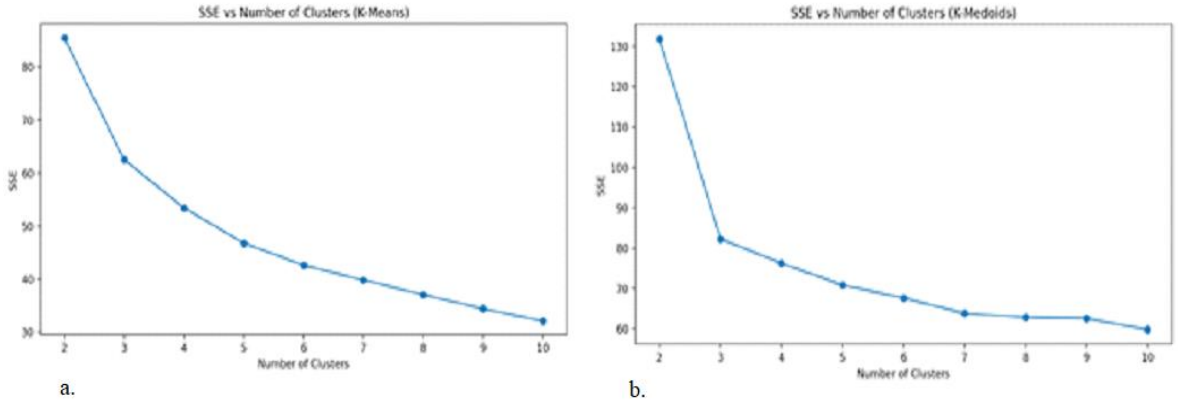

Figure 2. SSE Performance Comparison a.K-Means; b. K-Medoids.

Tabel 4. Performance measurement of feature importance

Criteria	Mean	Standart Deviasi	Chi Square
Type of Cattle	0.4708	0.5001	1.1521%
Omset	0.3920	0.4292	3.7218%
Age of Cattle	0.5632	0.3799	4.4989%
Weight	0.4781	0.0231	0.7854%
Price of Meat	0.8092	0.2032	4.5332%
Length of Cattle	0.1901	0.3886	2.9220%
Vaccine	0.2028	0.0598	2.4669%
Type of feed	0.3754	0.0091	2.2186%
Types of yeast	0.2387	0.2003	1.1575%
Long drying time	0.2490	0.4807	3.7257%
Storage time	0.3293	0.5276	5.4930%
Types of yeast	0.2276	0.3776	0.7800%

The measurement GS-KMBSC with cluster 3-th will analyze the importance of attributes using standard deviation and mean in Table 4 shows SD measures the extent to which data is spread around its average value. The higher the standard deviation, the greater the variation or

heterogeneity of the data. Conversely, a low standard deviation indicates that the data tends to be closer or more homogeneous around its mean value.

Performance measurement based on importance feature representation uses chi-square to calculate relevant feature selection to measure the level of importance of features for building the model. In cluster 3, the chi-square test tests the relationship between features for the best number of k clusters and N data points. For example, Ci is the ith cluster where xi is the ith point in the ith cluster, wich shown the most dominant attribute in storage time. The grouping pattern for each cluster was obtained using the maximum standard deviation value of 0.5276 and the mean 0.3293 for one attribute type of cattle in the dominant cluster. The higher the standard deviation value, the more variation value. Meanwhile, the average value shows that the data is close to stationary, has similarities, and is a relatively stagnant value. Furthermore, the value based on percentage with chi-square shows the most dominant attribute of 5.4930% for storage time, indicating the attribute that has the most influence on the development of beef processing SMEs.

5. CONCLUSIONS

The K-Medoids hyperparameter grouping method with centroid refinement with binary search can be used to group data that do not have labels or previous class information. The research on beef processing data without labels previously obtained 3 clusters in DBI measurements. The clustering process involves some stages, including data preprocessing, using the label encoding method to convert categorical data into numerical data, data imputation using the mean method to fill in empty values, and finally, data normalization using the min-max method to ensure the data has a uniform scale. In the grouping process using K-Medoids, evaluation is carried out using DBI and SSE to measure the quality of both. Cluster analysis GS-K-Medoids measurements with the lowest DBI in cluster 3 were worth 0.1021, in the experimental range with the number of clusters from 2 to 10. So, the found that the lowest SSE results occurred in cluster 10, with an SSE value of 32.0970 for the K-Medoids method, lower than Traditional K-Means is worth 50.9282. Thus, the analysis of each cluster in both methods shows that the K-Medoids method is more optimal because it fits categorical data. Meanwhile, the analysis of attribute importance at c=3 shows that

the top order of attributes shows that the best attributes occur in the criteria of storage time and the second in the price of meat. Each grouping pattern Clusters were obtained based on determining the quality of beef processing. Dominant clusters were found in local cattle because they have dense fiber. Based on the weaknesses of the clustering method, it is very dependent on centroid selection and missing data because it affects clustering performance. So, computational methods for selecting clustering hyperparameters and correcting missing value data are highly recommended.

FUNDING

We thank all our supporters for The Research and Community Service in Trunojoyo Madura of the University through the National Collaborative Research grant program, which supported the funding budget 2023. We also thank the Islamic University of Malang, which has supported this research with its collaboration team and all experimental facilities so it can be carried out well.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] B.W. Penning, W.M. Snelling, M.J. Woodward-greene, machine learning in the assessment of meat quality, IT Prof. 22 (2020), 39-41. https://doi.org/10.1109/mitp.2020.2986123.

[2] S.B. Siswijono, P.S. Winarto, R. Prafitri, Strategy for improving production performance and preservation of madura cattle, IOP Conf. Ser.: Earth Environ. Sci. 478 (2020), 012072. https://doi.org/10.1088/1755-1315/478/1/012072.

[3] M. Gancarczyk, J. Gancarczyk, Proactive international strategies of cluster SMEs, Eur. Manage. J. 36 (2018), 59-70. https://doi.org/10.1016/j.emj.2017.03.002.

[4] R.M.F. Silveira, D.F. Lima, B.V. Camelo, et al. Machine learning applied to understand perceptions, habits and preferences of lamb meat consumers in the Brazilian semi-arid region, Small Ruminant Res. 227 (2023), 107088. https://doi.org/10.1016/j.smallrumres.2023.107088.

[5] A. Ahmad, L. Dey, A k-mean clustering algorithm for mixed numeric and categorical data, Data Knowl. Eng. 63 (2007), 503-527. https://doi.org/10.1016/j.datak.2007.03.016.

[6] A. Viloria, O.B. Pineda Lezama, Improvements for determining the number of clusters in k-means for innovation databases in SMEs, Procedia Computer Sci. 151 (2019), 1201-1206. https://doi.org/10.1016/j.procs.2019.04.172.

[7] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Mining Knowl. Discov. 2 (1998), 283-304. https://doi.org/10.1023/a:1009769707641.

[8] S. Reyya, M. Puspha, B. Priyanka, et al. Increasing comparison performance using k-harmonic mean, Int. J. Manage. Inform. Technol. Eng. 2 (2014), 11-18.

[9] S.F. Abushilah, R.H. Abbas, Performance Evaluation of Some Clustering Algorithms under Different Validity Indices, Math. Model. Eng. Probl. 10 (2023), 1271-1280. https://doi.org/10.18280/mmep.100420.

[10] Q. Li, X. Liu, A K-medoids Clustering Algorithm with Initial Centers Optimized by a P System, in: Q. Zu, B. Hu, N. Gu, S. Seng (Eds.), Human Centered Computing, Springer International Publishing, Cham, 2015: pp. 488–500. https://doi.org/10.1007/978-3-319-15554-8_40.

[11] R. Vankayalapati, K.B. Ghutugade, R. Vannapuram, et al. K-means algorithm for clustering of learners performance levels using machine learning techniques, Rev. d'Intell. Artif. 35 (2021), 99-104. https://doi.org/10.18280/ria.350112.

[12] R.S. Pontoh, S. Mulyani, S. Zhahira, et al. Mapping Indonesian potential fishing zone using hierarchical and non-hierarchical clustering, Commun. Math. Biol. Neurosci. 2023 (2023), 82. https://doi.org/10.28919/cmbn/8088.

[13] A.K. Abdalameer, M. Alswaitti, A.A. Alsudani, et al. A new validity clustering index-based on finding new centroid positions using the mean of clustered data to determine the optimum number of clusters, Expert Syst. Appl. 191 (2022), 116329. https://doi.org/10.1016/j.eswa.2021.116329.

[14] Y. Zhu, F. Wang, X. Shan, et al. K-medoids clustering based on MapReduce and optimal search of medoids, in: 2014 9th International Conference on Computer Science & Education, IEEE, Vancouver, BC, Canada, 2014: pp. 573–577. https://doi.org/10.1109/ICCSE.2014.6926527.

[15] H.S. Park, C.H. Jun, A simple and fast algorithm for K-medoids clustering, Expert Syst. Appl. 36 (2009), 3336–3341. https://doi.org/10.1016/j.eswa.2008.01.039.

[16] Z. Chen, G. Li, J. He, et al. A new parallel adaptive structural reliability analysis method based on importance sampling and K-medoids clustering, Reliab. Eng. Syst. Safe. 218 (2022), 108124. https://doi.org/10.1016/j.ress.2021.108124.

[17] A.V. Ushakov, I. Vasilyev, Near-optimal large-scale k-medoids clustering, Inform. Sci. 545 (2021), 344-362. https://doi.org/10.1016/j.ins.2020.08.121.

[18] H. Song, J.G. Lee, W.S. Han, PAMAE: Parallel k-medoids clustering with high accuracy and efficiency, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Halifax NS Canada, 2017: pp. 1087-1096. https://doi.org/10.1145/3097983.3098098.

[19] S. Harikumar, P.V. Surya, K-medoid clustering for heterogeneous datasets, Procedia Computer Sci. 70 (2015), 226-237. https://doi.org/10.1016/j.procs.2015.10.077.

[20] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, Neurocomputing. 415 (2020), 295-316. https://doi.org/10.1016/j.neucom.2020.07.061.

[21] H.A. Fayed, A.F. Atiya, Speed up grid-search for parameter selection of support vector machines, Appl. Soft Comput. 80 (2019), 202-210. https://doi.org/10.1016/j.asoc.2019.03.037.

[22] M. Ogunsanya, J. Isichei, S. Desai, Grid search hyperparameter tuning in additive manufacturing processes, Manufact. Lett. 35 (2023), 1031-1042. https://doi.org/10.1016/j.mfglet.2023.08.056.

[23] W. Sheng, X. Liu, A genetic k-medoids clustering algorithm, J. Heuristics. 12 (2006), 447-466. https://doi.org/10.1007/s10732-006-7284-z.

[24] F. Batool, C. Hennig, Clustering with the average silhouette width, Comput. Stat. Data Anal. 158 (2021), 107190. https://doi.org/10.1016/j.csda.2021.107190.

[25] A. Hatamlou, In search of optimal centroids on data clustering using a binary search algorithm, Pattern Recogn. Lett. 33 (2012), 1756-1760. https://doi.org/10.1016/j.patrec.2012.06.008.

[26] P. Arora, Deepali, S. Varshney, Analysis of k-means and k-medoids algorithm for big data, Procedia Computer Sci. 78 (2016), 507-512. https://doi.org/10.1016/j.procs.2016.02.095.

[27] D. Sun, H. Fei, Q. Li, A bisecting k-medoids clustering algorithm based on cloud model, IFAC-PapersOnLine. 51 (2018), 308-315. https://doi.org/10.1016/j.ifacol.2018.08.301.

[28] N. Santoro, J.B. Sidney, Interpolation-binary search, Inform. Process. Lett. 20 (1985), 179-181. https://doi.org/10.1016/0020-0190(85)90046-8.

[29] A.S. Mohammed, Ş.E. Amrahov, F.V. Çelebi, Interpolated binary search: An efficient hybrid search algorithm on ordered datasets, Eng. Sci. Technol. Int. J. 24 (2021), 1072-1079. https://doi.org/10.1016/j.jestch.2021.02.009.

[30] T. Xu, J. Jiang, A Graph Adaptive Density Peaks Clustering algorithm for automatic centroid selection and effective aggregation, Expert Syst. Appl. 195 (2022), 116539. https://doi.org/10.1016/j.eswa.2022.116539.

[31] A.E. Jacob, N. Ashodariya, A. Dhongade, Hybrid search algorithm: Combined linear and binary search algorithm, in: 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), IEEE, Chennai, 2017: pp. 1543–1547. https://doi.org/10.1109/ICECDS.2017.8389704.

[32] H. Lorentz, O.P. Hilmola, J. Malmsten, et al. Cluster analysis application for understanding SME manufacturing strategies, Expert Syst. Appl. 66 (2016), 176-188. https://doi.org/10.1016/j.eswa.2016.09.016.

[33] H. Hu, J. Liu, X. Zhang, et al. An effective and adaptable k-means algorithm for big data cluster analysis, Pattern Recogn. 139 (2023), 109404. https://doi.org/10.1016/j.patcog.2023.109404.

[34] J. Saha, J. Mukherjee, CNAK: Cluster number assisted K-means, Pattern Recogn. 110 (2021), 107625. https://doi.org/10.1016/j.patcog.2020.107625.

[35] S. Zhou, Z. Xu, A novel internal validity index based on the cluster centre and the nearest neighbour cluster, Appl. Soft Comput. 71 (2018), 78-88. https://doi.org/10.1016/j.asoc.2018.06.033.

[36] M. Matarazzo, L. Penco, G. Profumo, et al. Digital transformation and customer value creation in Made in Italy SMEs: A dynamic capabilities perspective, J. Bus. Res. 123 (2021), 642-656. https://doi.org/10.1016/j.jbusres.2020.10.033.

[37] T. Taurino, A Cluster Reference Framework for Analyzing Sustainability of SME Clusters, Procedia CIRP. 30 (2015), 132-137. https://doi.org/10.1016/j.procir.2015.02.138.

[38] M. Afsharian, P. Bogetoft, Limiting flexibility in nonparametric efficiency evaluations: An ex post k-centroid clustering approach, Eur. J. Oper. Res. 311 (2023), 633-647. https://doi.org/10.1016/j.ejor.2023.05.020.