



Outlier Handling in Clustering: A Comparative Experiment of K-Means, Robust Trimmed K-Means, and K-Means Least Trimmed Squared

Tricia Estella¹, Nadzla Andrita Intan Ghayatrie², and Antoni Wibowo³

^{1,2,3} Master of Information Technology, BINUS Graduate Program, BINUS University, Jakarta, Indonesia

E-mail address: tricia.estella@binus.ac.id, nadzla.ghayatrie@binus.ac.id, anwibowo@binus.edu

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: The presence of outliers in data often leads to unsatisfactory modeling outcomes, especially when employing clustering algorithms for population segmentation and behavioral analysis. While various outlier-resilient clustering algorithms like DBSCAN, LDOF, t-SNE, and others exist, one of the most renowned algorithms, k-Means, still faces challenges in effectively handling outliers. This journal proposes an optimization of the k-Means algorithm resilient to outliers by incorporating the Least Trimmed Square technique as post-processing, referred to as k-Means LTS. The outlier trimming process occurs after the grouping process, allowing trimming within each cluster. This algorithm will be compared with ordinary k-Means and Robust Trimmed k-Means, as known as RTKM, both employing outlier trimming. The comparison of these three algorithms will consider performance metrics, clustering results, and running time. The contribution of this research lies in the enhanced optimality of k-Means LTS algorithm, outperforming the other two algorithms across all comparison parameters. By utilizing this algorithm, the presence of outliers within each cluster can be more easily explained, and the running time is notably shorter compared to RTKM. As a result, the proposed algorithm of k-Means LTS consistently proves to work better than ordinary k-Means and RTKM when implemented across ten datasets of varying types.

Keywords: Clustering; Least Trimmed Squares; K-Means; Robust clustering; Noisy data; Outliers

1. INTRODUCTION

Outliers in data often pose a significant challenge in data analysis and model development. An outlier is a data point that deviates significantly from the established or expected data norms [1]. In essence, it stands out prominently from the rest of the data. For instance, consider a dataset of student grades: 60, 60.5, 62, 61, 63, 60, and 0.01. The value 0.01 in this data is an outlier and can adversely impact the modeling process, leading to suboptimal results. Numerous journals discuss methods for detecting and handling outliers in data. Presently, many clustering algorithms, such as DBSCAN, LDOF, t-SNE, K-Medoids, and others, are employed to simultaneously address outliers. Each algorithm has its approach.

Clustering modeling is a powerful technique for directly identifying outliers. Essentially, clustering groups data with similar characteristics into clusters. Consequently, if a data point appears significantly distant

from its assigned cluster, it is deemed an outlier. Although additional functions are required to handle outliers, models like K-Means may not perform optimally but can still identify outliers effectively [2].

This article proposes a novel algorithm by implementing the Least Trimmed Square (LTS) algorithm after performing clustering using K-Means. Outlier trimming involves sorting the distances of each data point from its centroid. The optimal trimming percentage is determined by searching for the highest silhouette value. If a cluster exhibits a high silhouette value, that percentage is selected. While existing literature employs LTS as inspiration for trimmed K-Means algorithms [3], [4], [5], which trim the farthest points during centroid and cluster calculations, the use of LTS in the proposed method has not been explored before. Therefore, we will compare the use of K-Means as a control algorithm with trimmed K-Means to handle datasets with significant outliers, and we will also compare it with the novel algorithm we introduce, K-Means LTS. Through

experiments, result analysis, and time performance evaluations, this article aims to demonstrate and discover a more effective algorithm for forming clusters in the presence of outliers.

This article is structured as follows: Section 2 explores Related Works, providing a review of existing literature to establish context and identify research gaps. Section 3 details the chosen clustering algorithm, evaluation metrics, and other methodologies employed. In Section 4, Experiment & Analysis, we present datasets, preprocessing steps, and optimization details. Section 5, Result & Discussions, showcases outcomes through evaluation metrics and visualizations, accompanied by an in-depth analysis. Finally, Section 6, Conclusion & Future Works, summarizes findings, discusses implications, and suggests directions for future research, ensuring a comprehensive exploration of clustering methodologies and their applications.

2. RELATED WORKS

In the journal [6], the researchers extensively discuss the modification of the K-Means clustering model, making it more resilient to outliers, referred to as Robust Trimmed K-Means (RTKM). Another journal from 2019 [5] mentions that Trimmed C-Means and Trimmed K-Means implement LTS criteria within them. This journal implements LTS only conceptually in its trimming, not incorporating the LTS method as a pre-processing step, as seen in the previous journal. Ikotun, in a journal utilizing the Systematic Literature Review (SLR) method summarizing K-Means Clustering-related journals [7], notes that many journals discuss the workings of Robust K-Means. Some K-Means types address outliers by discarding them during the iteration of cluster centroid determination. Ikotun also explains journals that explore K-Means by combining Tukey Rule and a new distance metric formula [8]. The algorithm is modified to eliminate outliers before finding cluster centroids, resulting in improved accuracy and convergence.

There is another journal discussing the use of single-linked clustering algorithms focused on identifying elongated clusters and ultimately finding inliers reflecting majority patterns or patterns matching the data [9]. Moreover, this journal employs Least Square (LS) and Least Trimmed Square (LTS) as comparative estimators. Trimmed K-Means is an algorithm inspired by the trimming concept in LTS and Minimum Covariance Determinant (MCD) according to Rousseeuw's journal [10]. It explains that trimming in K-Means minimizes the sum of squared distances between observation objects (subsets) and group averages. The algorithm broadly utilizes the concept of C-steps for each iteration, similar to FastMCD. Through these previous journals, they inspire us to explore new algorithms to identify outliers in each cluster formed in K-Means, making the clustering results

more robust without requiring high computing resources like existing algorithms

3. METHODS

This section succinctly outlines the chosen clustering algorithm, evaluation metrics, and any supplementary techniques employed in the study. It offers a non-technical overview of the technical approach adopted for assessing algorithm performance, providing readers with a clear understanding of the research methodology.

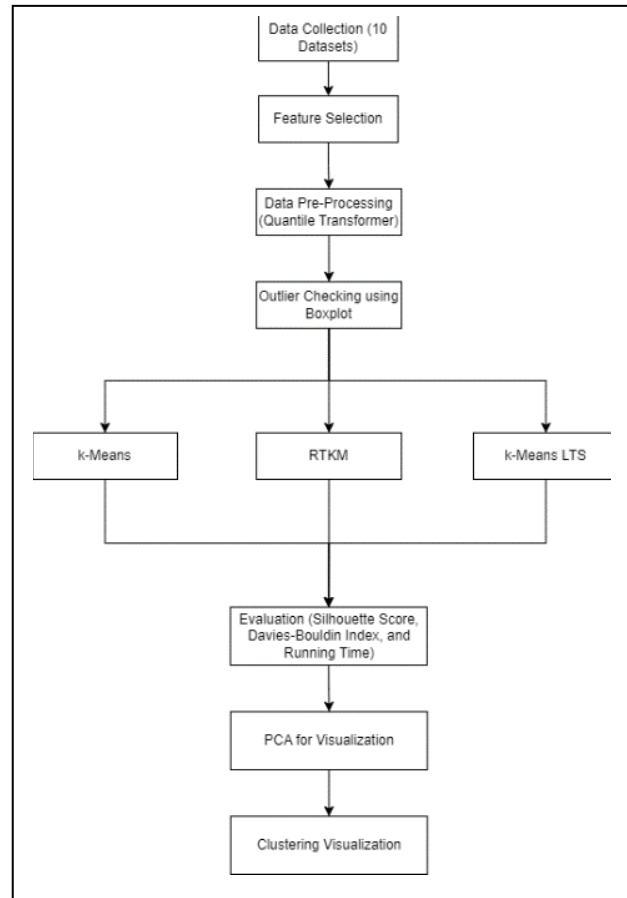


Figure 1. Research Framework

A. K-Means

The K-Means Clustering algorithm is the most renowned algorithm in unsupervised learning. Not only is it fast in convergence, but K-Means is also easy to understand and performs well on large datasets. K-Means clustering is categorized as a partitioning algorithm that divides data into specific groups or clusters [7]. The partitioning algorithm determines the number of groups from the beginning and iteratively relocates between groups to become more centralized or converge [11]. The objective of this algorithm is to minimize the average Euclidean distance of each sample from the cluster center (centroid) [12], where Euclidean distance is used when

assigning a data point (x) to a cluster, considering the distance between the data point (x) and the cluster center (c).

$$d(x, c) = \sqrt{\sum_{i=1}^n (x_i - c_i)^2} \quad (1)$$

By using the K-Means Clustering algorithm, various information can be obtained, such as web keyword sources [13], image segmentation [14], customer segmentation identification in a company [15], and much more. However, a drawback of this renowned algorithm is its inability to handle datasets with a large number of outliers. Therefore, the K-Means clustering results will serve as the control in this article, to be compared with the modified version of the K-Means algorithms.

B. Trimmed K-Means

The Least Trimmed Squares (LTS) estimator is a modification of the Least Square (LS) estimator in linear regression, designed to be robust against outliers by finding regression coefficients that minimize the sum of squares (the difference between observed and model-predicted values). Least Trimmed Squares (LTS) is a concept proposed by Rousseeuw in robust regression modeling susceptible to outliers as an estimator for linear coefficient [16]. Very similar to LS, the only difference in LTS is that the largest squared residuals are not used in summation, preventing the model from being affected by outliers.

$$\text{Minimize } \sum_{i=1}^h (r^2)_{i:n}$$

$$\text{Where } (r^2)_{1:n} \leq \dots \leq (r^2)_{n:n} \quad (2)$$

The primary application of LTS is in linear regression models, where the squared residuals of predictions and actual values are sorted in ascending order, and trimming is performed on the largest squared residuals of the model. By estimating a robust linear model using LTS, data points with the largest residuals are considered outliers. LTS has been implemented in clustering algorithms, both in hard clustering and fuzzy clustering. LTS inspires these algorithms to trim the farthest data points (Trimming Approach). The Trimmed K-Means algorithm is a modification of the K-Means algorithm, implementing the concept of Least Trimmed Squares in each iteration of the cluster search until the convergence condition is met. The trimming concept from LTS inspires the trimming process in Trimmed K-Means, in the univariate context,

demonstrating their similarity in seeking robust locations against outliers [17]. In this algorithm, determining clusters involves not only least squares (LS) but also cutting unnecessary values (trimming approach) [4]. Therefore, the main step that distinguishes Robust Trimmed K-Means is the selection of a separate subset (trimming). In this step, points with the largest residuals (which may be outliers) are removed from the dataset or ignored in the next iteration. This concept is inspired by LTS. With the subset of data after trimming, the algorithm iterates between points again, updating cluster labels and cluster centers based on the remaining data until convergence.

In this study, we will utilize the Robust Trimmed K-Means (RTKM) algorithm [18], for which the code is accessible on GitHub. This algorithm is based on a robust relaxed formulation of the weighted k-means algorithm, where the classification weight matrix can take on a range of values within [0, 1], as opposed to being restricted to the binary set {0, 1}. This relaxation provides a method for monitoring the degree of membership of a point to each cluster during each iteration, eliminating the need for explicitly defining a measure of "outlierness."

C. K-Means LTS

In theory, LTS can be used as a preprocessing algorithm where the LTS estimator is employed to detect outliers before running the clustering algorithm. This process does not modify the clustering algorithm but enhances its robustness by processing outliers beforehand. Therefore, the application of LTS in clustering modeling can make the model more resilient to outliers, a preprocessing step that will be discussed in this article.

The proposed method shares similarities with the trimmed K-Means algorithm or RTKM algorithm, where the LTS concept inspires and is applied to the clustering method. LTS inspires the proposed preprocessing in this article by sorting the clustering result dataset per cluster and trimming the largest data points estimated as outliers in the K-Means algorithm. The sorting of residuals in the K-Means LTS algorithm will be performed by arranging the farthest distances to the formed cluster centroid. Subsequently, trimming the farthest points will be executed based on a certain percentage. The search for the optimal trimming percentage will be done in an unsupervised manner, eliminating the need for multiple trials [3]. The algorithm for determining the optimal trimming percentage is based on the largest evaluation metrics result, namely the silhouette score.



TABLE I. K-MEANS LEAST TRIMMED SQUARES ALGORITHM

Algorithm 1 K-Means LTS

```

1 procedure K_Means_LTS(data, n_cluster, n_percent):
2   centroids = initialize_centroids(data, n_cluster)
3   assign_clusters(data, centroids)
4   calculate_distances(data, centroids)
5   silhouette_scores = []
6   for percent in range(1, n_percent):
7     sorted_data = sort_data_by_distance(data)
8     n_rows_to_trim = int(n_percent / 100 * len(sorted_data))
9     inliers_data = sorted_data.tail(n_rows_to_trim)
10    silhouette_score = calculate_silhouette(inliers_data)
11    silhouette_scores.append((percent, silhouette_score))
12  best_percent = find_best_percent(silhouette_scores)
11  inliers_result = trim_data(data, best_percent)
12  return inliers_result

```

D. Evaluation Metrics

Evaluation metrics provide a systematic and quantitative means to measure the quality of clustering results, guiding researchers and practitioners in selecting the most suitable algorithm for their specific dataset and objectives, as well as quantifying the performance and reliability of the generated clusters. We used several evaluation metrics available to score our clustering result, which are:

1) *Silhouette Score*: The evaluation of clustering models encompasses various methods, with one commonly used approach being the Silhouette score. Notably, the Silhouette score stands out due to its independence from training set values, making it well-suited for clustering models. This score is employed to assess the clustering algorithm's effectiveness, considering both inter-cluster separation and intra-cluster cohesion. Negative Silhouette values indicate suboptimal object placement, while positive values signify improved placement [19]. The Silhouette function is expressed as:

$$S = \frac{b-a}{\max(a,b)} \quad (3)$$

Where 'a' denotes the average distance from a data point to all other points within the same cluster, while 'b' represents the minimum average distance from the data point to all other points in any alternative cluster.

2) *Davies-Bouldin Index (DBI)*: David L. Davies and Donald W. Bouldin introduced the Davies-Bouldin Index (DBI) as a method for assessing clusters, specifically focusing on internal cluster evaluation. This index evaluates the quality of cluster results based on both their quantity and proximity in grouping methods, considering cohesion (the sum of data proximity to the cluster center point) and separation (the distance between cluster center

points). The primary objective is to maximize inter-cluster distance while minimizing intra-cluster distance, highlighting differences between clusters and indicating high characteristic similarity within clusters. The DBI serves as a metric for cluster validity, with a lower value indicating successful, well-separated, and compact clusters, while higher values suggest inadequate separation and compactness [20].

3) *Elbow Method*: The Elbow Method is a graphical method for finding the optimal number of clusters (K) in K-Means clustering. The method involves finding the within-cluster sum of square (WCSS), which is the sum of the square distance between points in a cluster and the cluster centroid. WCSS (Within-Cluster Sum-of-Squares) gauges the variance within each cluster. Lower overall WCSS values indicate better clustering. The WCSS values are plotted against the different values of K, and the optimal K value is the point at which the graph forms an elbow [21]. The point at which this elbow occurs is considered the optimal number of clusters, as adding more clusters beyond this point does not significantly improve the model's performance. This method provides an intuitive and visual way to determine a reasonable number of clusters for a given dataset, aiding in the decision-making process when using clustering algorithms.

E. Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique commonly used in data analysis and machine learning. Its primary goal is to transform a dataset with potentially high-dimensional features into a new set of orthogonal (uncorrelated) variables called principal components. These components capture the maximum variance present in the original data, allowing for a more compact representation [22]. By

selecting a subset of principal components, one can reduce the dimensionality of the data while retaining the essential information. PCA is widely employed for feature extraction, noise reduction, and visualization, contributing to improved efficiency and interpretability in various analytical tasks.

This algorithm will be used for visualization purposes, where cluster results will be easier to interpret within 2 dimensions (features).

4. EXPERIMENT AND ANALYSIS

The selected methodology is applied in a practical context, presenting a thorough examination of the datasets used in our experiments. We outline the datasets used on this experiment, then a step-by-step process of preprocessing, optimizations, and other crucial procedures. The structured approach aims to provide a comprehensive view of the experimental design, elucidating the rationale behind key decisions made throughout the research. This experiment was conducted using an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHZ with 16.0 GB of memory.

A. Datasets

In this section, we introduce the datasets utilized in our comparative experiment, aiming to evaluate the effectiveness of three distinct clustering methodologies—K-Means, Trimmed K-Means, and K-Means Least Trimmed Squares (LTS)—in the context of outlier detection. Our experiment encompasses a diverse collection of 10 datasets gathered from various sources, including Kaggle, to ensure a comprehensive evaluation. The characteristics and origins of each dataset are detailed in the table below, providing valuable insights into the varied nature of the data and offering a robust foundation for assessing the performance of the clustering algorithms in the subsequent sections of this study.

TABLE II. DATASETS USED ON THIS EXPERIMENT

No	Name	Num of Features	Num of Data	Source
1	Air Company Customer Info	3	62988	Kaggle
2	NSE Banking Sector	15	41231	Kaggle
3	CC General	18	8950	Kaggle
4	Sonar	61	208	UCI ML
5	Wholesale Customers	8	440	Kaggle
6	Obesity Data	17	2111	Kaggle
7	Diabetes	9	768	Kaggle
8	Nurse Stress Prediction Wearable Sensors	6	2000	Kaggle
9	Marketing Campaign	29	2240	Kaggle
10	Bank Customer Segmentation	9	5304	Kaggle

B. Preprocessing

For achieving optimal cluster results across the three algorithms employed, a preprocessing step is essential for each dataset under consideration. The preprocessing steps

are tailored to the characteristics of the respective datasets. A common preprocessing technique we employ involves handling null data and transforming the data to achieve a well-distributed format without altering the data values. Specifically, we utilize quantile transformation, a technique that maps the original data distribution to a predefined target distribution, often a standard normal distribution Bogner et al. (2012). This process assigns each data point its corresponding quantile in the target distribution, ensuring that the transformed data adheres to the desired distribution. Quantile transformation proves advantageous in preprocessing datasets for clustering by effectively mitigating the impact of outliers and non-gaussian distributions. This adaptation makes the data more suitable for algorithms that assume normality or exhibit sensitivity to outliers. The transformation to a standard distribution enhances the robustness and performance of clustering algorithms, enabling them to operate more effectively and consistently across diverse datasets.

C. Optimizations

To create optimal cluster results for each dataset using the three tested algorithms, we devised optimization steps to determine the best number of clusters and the optimal trimmed percentage for the RTKM and K-Means LTS algorithms.

1) *Number of Clusters*: To determine the best number of clusters (K), we employ the elbow method for each K value ranging from two to eight, used to find the optimal Within-Cluster Sum of Squares (WCSS) for the regular K-Means algorithm. In cases where the elbow point is not clearly visible, we refer to the silhouette score for the K value in K-Means for the tested dataset. This unsupervised finding of the best K is then utilized for all three algorithms under investigation: K-Means, RTKM, and the proposed K-Means LTS in this article.

2) *Optimal Trimmed Percentage*: After determining the best number of clusters (K), we proceeded with clustering for the K-Means, RTKM, and K-Means LTS algorithms. For the RTKM and K-Means LTS algorithms, which require a trimming percentage, we searched for the optimal percentage by iteratively seeking the best silhouette score. The percentage range used starts from 5% up to 30%, with an increment value of five percent for each iteration. However, we observed that the optimal trimmed percentages for both RTKM and K-Means LTS algorithms consistently yielded the highest silhouette scores at 30%. Increasing the percentage of outlier data removal in the algorithms improves the silhouette score by eliminating the influence of isolated points. This results in the formation of more homogeneous clusters, contributing to a higher silhouette score.



5. RESULT AND DISCUSSION

In this section, we delve into the outcomes of the comparative experiments of the proposed K-Means, RTKM, and K-Means LTS algorithms. The evaluation is based on silhouette score, David Bouldin Index, and cluster visualizations, providing insights into how these methods respond to outliers in the clustering data.

A. Silhouette Score

The obtained silhouette values across each dataset for the three models reveal notable distinctions. Specifically, the silhouette values derived from the k-Means model are markedly smaller in comparison to the other two models, namely k-Means LTS and RTKM. A detailed examination of Tables III, IV, and V underscores that the silhouette values attained from the k-Means LTS model outperform those of k-Means and RTKM across all datasets. This observation suggests that the k-Means LTS algorithm operates optimally, leading to the formation of well-defined and distinct clusters. The superior silhouette values for k-Means LTS underscore its efficacy in achieving optimal cluster separation, reflecting its robust performance in the context of the analyzed datasets.

B. Davies-Bouldin Index

Similar to the obtained Silhouette values, the Davies-Bouldin Index differs among the algorithms. A closer examination of the evaluation matrices in Tables IV and V reveals that both algorithms, k-Means LTS and RTKM, exhibit larger indices for datasets 3, 4, 6, and 9. These particular datasets display suboptimal cluster divisions, signaling the need for improved analytical techniques and preprocessing methods. Nevertheless, it is noteworthy that the index generated by k-Means LTS is consistently lower than that of RTKM, as evident in the evaluation across all 10 datasets. This suggests that k-Means LTS excels not only in Silhouette values but also in achieving lower Davies-Bouldin Indices, affirming its superior performance across the diverse dataset scenarios.

Here is the evaluation metrics table depicting the overall results of the experiments, including the outcomes of the best number of clusters (K) and the optimal trimmed percentage (%) obtained in the previous step:

TABLE III. EVALUATION METRICS FOR K-MEANS

No	Name	Best K	Silhouette	DBI
1	Air Company Customer Info	2	0.6908	0.4076
2	NSE Banking Sector	2	0.3644	1.0033
3	CC General	7	0.3724	1.2426
4	Sonar	3	0.1165	2.2625
5	Wholesale Customers	4	0.6377	0.5807
6	Obesity Data	4	0.2876	1.6111
7	Diabetes	3	0.4020	1.1342
8	Nurse Stress Prediction Wearable Sensors	7	0.4311	0.8877
9	Marketing Campaign	2	0.4830	1.4402
10	Bank Customer Segmentation	2	0.5841	0.6127

TABLE IV. EVALUATION METRICS FOR RTKM

No	Name	Optimal %	Silhouette	DBI
1	Air Company Customer Info	30	0.7516	0.3572
2	NSE Banking Sector	20	0.3965	0.9639
3	CC General	30	0.3866	1.0826
4	Sonar	15	0.1223	2.4719
5	Wholesale Customers	25	0.7108	0.4354
6	Obesity Data	25	0.3923	1.1222
7	Diabetes	25	0.5664	0.6857
8	Nurse Stress Prediction Wearable Sensors	25	0.4458	0.7168
9	Marketing Campaign	15	0.5962	1.002
10	Bank Customer Segmentation	30	0.6416	0.5074

TABLE V. EVALUATION METRICS FOR K-MEANS LTS

No	Name	Optimal %	Silhouette	DBI
1	Air Company Customer Info	30	0.7844	0.2839
2	NSE Banking Sector	30	0.4253	0.8902
3	CC General	25	0.4121	1.0762
4	Sonar	30	0.1275	2.1569
5	Wholesale Customers	30	0.7250	0.4735
6	Obesity Data	30	0.3811	1.3621
7	Diabetes	30	0.5474	0.7305
8	Nurse Stress Prediction Wearable Sensors	30	0.5634	0.5730
9	Marketing Campaign	5	0.5096	1.3622
10	Bank Customer Segmentation	30	0.6672	0.4538

C. Visualization

We will present multiple examples of clustering results for K-Means, RTKM, and K-Means LTS through scatterplots. We will use the clustering results for three datasets: RFM, NSE Banking Sector, and Bank Customer Segmentation. We reduced the features of these three datasets to two dimensions via PCA for ease of visualization.

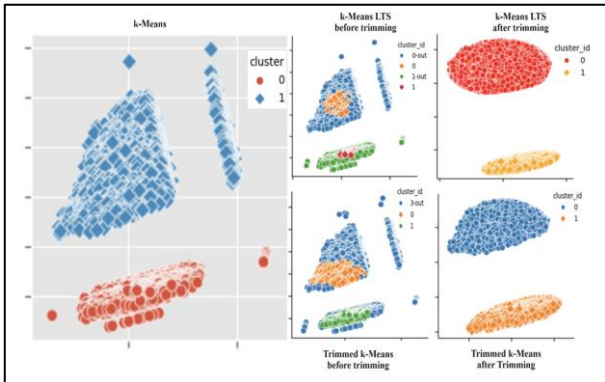


Figure 2. Visualization of k-Means, k-Means LTS, and RTKM results on the first dataset.

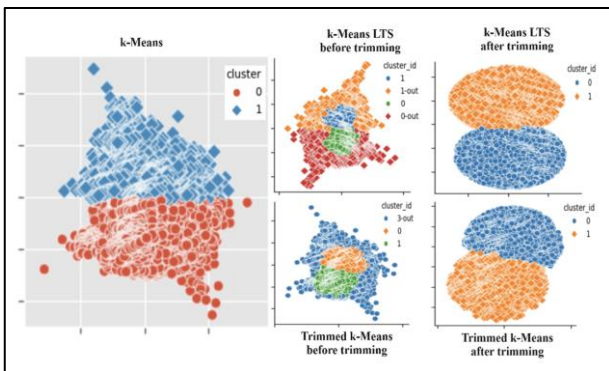


Figure 3. Visualization of k-Means, k-Means LTS, and RTKM results on the second dataset.

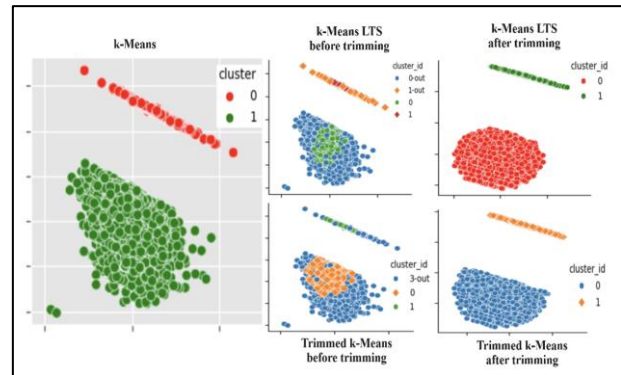


Figure 4. Visualization of k-Means, k-Means LTS, and RTKM results on the tenth dataset.

Based on these visualizations, the K-Means LTS algorithm effectively eliminates outliers in clustering, consistent with the optimal values of K and trimmed percentage. Trimming in K-Means LTS is conducted fairly for each cluster formed from the K-Means results, unlike RTKM, which iteratively trims by minimizing its objective function during the clustering process until convergence. Therefore, points identified as outliers by these two algorithms differ significantly.

However, in this experiment, we observed that the RTKM algorithm does not return the correct number of clusters according to the specified parameters. For example, using $K = 3$ for dataset RFM with the same parameters tested multiple times, the algorithm displays combinations of cluster numbers from 0 to 3 with outliers visible in the following visualization comparisons. Hence, it requires several attempts for the same K value until obtaining accurate results, thereby affecting processing time.

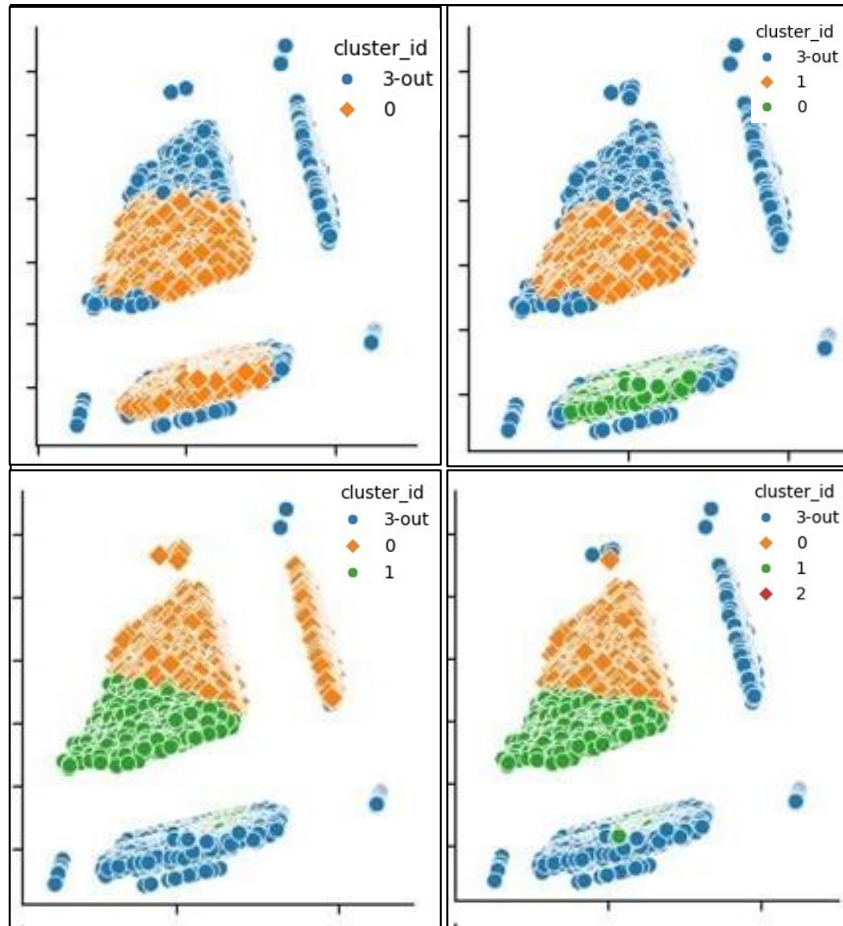


Figure 5. RTKM generated with K = 3 shows 1, 2, and 3 centroids with inconsistent cluster result.

In contrast, with the K-Means LTS algorithm, we can consistently return the correct number of clusters determined in the optimization phase. Additionally, in the visualization of K-Means LTS, we have the capability to identify the outliers that are in close proximity to specific clusters. This enables a reconsideration of these data points as outliers, unlike the RTKM algorithm, which groups all outliers into a single cluster. Therefore, the interpretability of the K-Means LTS algorithm is superior to the RTKM algorithm, as we consistently provide clustering results with the specified number of clusters with their outliers. Clearer visualization results through pairplots for all three algorithms for each dataset, including post-trimming and pre-trimming cluster results, can be accessed via the GitHub link provided at the end of the article.

6. CONCLUSION AND FUTURE WORKS

This study introduces an innovative clustering modeling algorithm that leverages Least Trimmed Square as a post-processing technique to effectively address outlier data within each cluster. Diverging from the methodology employed by RTKM, k-Means LTS operates by systematically removing outliers subsequent to the initial partitioning of data into distinct clusters. This distinctive approach enables the algorithm to elucidate the percentage of outliers trimmed within each cluster with greater granularity, offering a more specific and detailed insight into the outlier-handling process. The utilization of k-Means LTS thus contributes to an enhanced understanding of the algorithm's capability to manage and refine clusters by mitigating the impact of outliers in a targeted and specific manner.



Based on the conducted experiments, the K-Means LTS algorithm successfully eliminates outliers in the clustering results, with similar or better silhouette scores and Davies Bouldin Index with the RTKM algorithm. Furthermore, in terms of runtime, our algorithm competes effectively with the trimmed K-Means algorithm. The proposed algorithm consistently produces good clusters according to the best number of clusters and optimal percentage parameters determined during the optimization phase.

Furthermore, the future work directions outlined in [18] for determining two parameters (`n_cluster`, `percent_outliers`) have been identified through unsupervised application of the elbow method and silhouette score. However, further research is needed to determine the optimal value for `num_members` in the RTKM algorithm in an unsupervised manner, as this study primarily focuses on identifying outliers in hard-clustering K-Means for single-membership data. While we successfully determined the trimmed percentage by examining the best silhouette scores, there is room for future exploration where researchers can discover alternative metrics for determining optimal trimming, apart from relying solely on silhouette scores. Additionally, a more in-depth analysis of the results produced by the proposed K-Means LTS algorithm in this article could be pursued.

Code Availability:

<https://github.com/jptriciaestella/Skripsi-Kmeans-LTS>

ACKNOWLEDGMENT

We extend our gratitude to BINUS University for its unwavering support during the ideation and development of this research. This study wouldn't be possible without the resources and encouragement from the BINUS academic community.

REFERENCES

- [1] Smiti, A. (2020). A critical overview of outlier detection methods. *Computer Science Review*, 38, 100306.
- [2] Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8), 1295.
- [3] Kim, J., Krishnapuram, R., & Davé, R. (1996). Application of the least trimmed squares technique to prototype-based clustering. *Pattern Recognition Letters*, 17(6), 633-641.
- [4] Banerjee, A., & Davé, R. N. (2004, June). The feasible solution algorithm for fuzzy least trimmed squares clustering. In *IEEE Annual Meeting of the Fuzzy Information, 2004. Processing NAFIPS'04*. (Vol. 1, pp. 222-227). IEEE.
- [5] Ubukata, S. (2019). A unified approach for cluster-wise and general noise rejection approaches for k-means clustering. *PeerJ Computer Science*, 5, e238.
- [6] Dorabiala, O. (2023). Robust Approaches for Unsupervised Learning (Doctoral dissertation, University of Washington).
- [7] Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhajja, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622, 178-210.
- [8] Shrifan, N. H., Akbar, M. F., & Isa, N. A. M. (2022). An adaptive outlier removal aided k-means clustering algorithm. *Journal of King Saud University-Computer and Information Sciences*, 34(8), 6365-6376.
- [9] ISMAIL, N. F. The performance of clustering approach with robust mm-estimator for multiple outlier detection in linear regression.
- [10] Rousseeuw, P. J., & Hubert, M. (2018). Anomaly detection by robust statistics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(2), e1236.
- [11] Madhulatha, T. S. (2011, July). Comparison between k-means and k-medoids clustering algorithms. In *International Conference on Advances in Computing and Information Technology* (pp. 472-481). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [12] S. Misra, H. Li, and J. He, *Machine learning for subsurface characterization*, Gulf Professional Publishing, ISBN:9780128177365, 2019.
- [13] Li, L., Wang, J., & Li, X. (2020). Efficiency analysis of machine learning intelligent investment based on K-means algorithm. *Ieee Access*, 8, 147463-147470.
- [14] Hrosik, R. C., Tuba, E., Dolicanin, E., Jovanovic, R., & Tuba, M. (2019). Brain image segmentation based on firefly algorithm combined with k-means clustering. *Stud. Inform. Control*, 28(2), 167-176.
- [15] Li, Y., Chu, X., Tian, D., Feng, J., & Mu, W. (2021). Customer segmentation using K-means clustering and the adaptive particle swarm optimization algorithm. *Applied Soft Computing*, 113, 107924.
- [16] Rousseeuw, P. J., & Leroy, A. M. (2005). Robust regression and outlier detection. John Wiley & sons.
- [17] Cuesta-Albertos, J. A., Gordaliza, A., & Matrán, C. (1997). Trimmed k -means: an attempt to robustify quantizers. *The Annals of Statistics*, 25(2), 553-576.
- [18] Dorabiala, O., Kutz, J. N., & Aravkin, A. Y. (2022). Robust trimmed k-means. *Pattern Recognition Letters*, 161, 9-16.
- [19] Ogbuabor, G., & Ugwoke, F. N. (2018). Clustering algorithm for a healthcare dataset using silhouette score value. *Int. J. Comput. Sci. Inf. Technol*, 10(2), 27-37.
- [20] Mughnyanti, M., Efendi, S., & Zarlis, M. (2020). Analysis of determining centroid clustering x-means algorithm with davies-bouldin index evaluation. In *IOP Conference Series: Materials Science and Engineering* (Vol. 725, No. 1, p. 012128). IOP Publishing.
- [21] Bholowalia, P., & Kumar, A. (2014). EBK-means: A clustering technique based on elbow method and k-means in WSN. *International Journal of Computer Applications*, 105(9).
- [22] Maćkiewicz, A., & Ratajczak, W. (1993). Principal components analysis (PCA). *Computers & Geosciences*, 19(3), 303-342.
- [23] Bogner, K., Pappenberger, F., & Cloke, H. L. (2012). The normal quantile transformation and its application in a flood forecasting system. *Hydrology and Earth System Sciences*, 16(4), 1085-1094.



Tricia Estella currently is a graduate student enrolled in the Master of Information Technology program with a focus on Artificial Intelligence and Data Science at Bina Nusantara University in Indonesia. Her research interests lie in the applied fields of Machine Learning and Deep Learning across various

media platforms.



Nadzla Andrita Intan Ghayatri is a graduate student pursuing a Master of Information Technology degree with a focus on Data, Machine Learning, and Artificial Intelligence at Bina Nusantara University in Indonesia. Her research interests center around addressing gender gap issues through innovative applications

of data science, machine learning, and artificial intelligence.



Antoni Wibowo, a Member (M) of IAENG since 2012, earned his first degree in Applied Mathematics in 1995 and completed a master's degree in Computer Science in 2000. In 2003, he was granted a Japanese Government Scholarship (Monbukagakusho) to pursue Master and PhD programs in Systems and Information

Engineering at the University of Tsukuba, Japan. He successfully obtained his second master's degree in 2006 and completed his PhD in 2009. Dr. Wibowo's doctoral research focused on machine learning, operations research, multivariate statistical analysis, and mathematical programming, particularly in the development of nonlinear robust regressions using statistical learning theory.

Currently, Dr. Eng. Wibowo holds the position of Specialist Lecturer at Binus Graduate Program (Master in Computer Science) in Bina Nusantara University,

Indonesia. He remains actively engaged in research, focusing on machine learning, optimization, operations research, multivariate data analysis, data mining, computational intelligence, and artificial intelligence.