International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16, No.1 (Aug-24)

http://dx.doi.org/10.12785/ijcds/160175

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

Tricia Estella[1], Nadzla Andrita Intan Ghayatrie[2] and Antoni Wibowo[3]

[1,2,3]*Master of Information Technology, BINUS Graduate Program, BINUS University, Jakarta, Indonesia*

Received 11 Mar. 2024, Revised 16 May 2024, Accepted 17 May 2024, Published 15 Aug. 2024

Abstract: The presence of outliers in data often disrupts modeling results, especially in population clustering and behavioral analysis. Although there are various clustering algorithms that are robust to outliers, such as DBSCAN and t-SNE, K-Means still has challenges in dealing with them. This paper introduces an optimized K-Means LTS algorithm, which incorporates Least Trimmed Square technique to reduce outliers after clustering process. The outlier trimming process occurs after the clustering process, allowing for trimming within each cluster. This algorithm will be compared with K-Means and Robust Trimmed K-Means (RTKM), which both use outlier pruning. The comparison of these three algorithms will consider performance metrics using Silhouette Score and Davies-Bouldin Index, also the run time processes. As a result, K-Means LTS is consistently shown to perform better than K-Means and RTKM when implemented on ten various datasets. In the future, there may be further developments related to determining the best percentage for trimming outliers.

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

1. INTRODUCTION

Outliers present a notable challenge in data analysis and model development by significantly deviating from expected data norms [1]. They stand out distinctly within a dataset, impacting the modeling process and leading to suboptimal results. For instance, in a set of student grades like 60, 60.5, 62, 61, 63, 60, and 0.01, the value 0.01 is an outlier. Detecting and handling outliers is a topic widely discussed in various journals, with many clustering algorithms like DBSCAN, LDOF, t-SNE, and K-Medoids being utilized to address outliers alongside clustering tasks.

Clustering modeling is a robust method for directly pinpointing outliers by grouping similar data points into clusters. An outlier is then identified if it significantly differs from its cluster. Although models like K-Means may not handle outliers optimally, they can still effectively detect them [2]. Research in 2023 [3] has explored handling outliers in clustering using methods like the Robust Spectral Clustering Algorithm, which utilizes sub-Gaussian random variables to enhance outlier detection. However, such studies often involve a limited number of datasets for implementation.

This article proposes a novel algorithm by implementing the Least Trimmed Square (LTS) algorithm after performing clustering using K-Means, which will be called K-Means LTS. The LTS concept is applied by trimming the largest squared residuals on the cluster generation. This process involves sorting the distances of each data point from its centroid, and removing the data with furthest distance at an optimal percentage. The optimal percentage is determined through iterative experimentation. If a cluster trimmed at a certain percentage achieves the highest silhouette score during the iteration, that percentage is chosen. While existing literature employs LTS as inspiration for trimmed K-Means algorithms [4], [5], [6], which trim the farthest points during centroid and cluster calculations, the use of LTS in the proposed method has not been explored before. By employing outlier trimming based on centroid distance using LTS as a preprocessing step in the K-Means framework, K-Means LTS achieves robust clustering outcomes.

The method proposed here bears resemblance to the trimmed K-Means algorithm or RTKM, where the LTS concept serves as inspiration and is integrated into the clustering methodology. LTS guides the preprocessing steps by organizing the clustering results dataset per cluster and trimming the largest data points identified as outliers in the K-Means algorithm. The sorting of residuals in the K-Means LTS algorithm involves arranging the most distant points from the cluster centroid, followed by trimming these points based on a specified percentage. Therefore, we will

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

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 and Analysis, we present datasets, preprocessing steps, and optimization details. Section 5, Result and Discussions, showcases outcomes through evaluation metrics and visualizations, accompanied by an in-depth analysis. Finally, Section 6, Conclusion and 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 dissertation entitled “Robust Approaches for Unsupervised Learning”, 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) [7]. RTKM offers a structure for identifying outliers and clustering points concurrently within one objective function. Its algorithmic design allows flexibility for use with both single- and multi-membership data. RTKM sidesteps the challenges of explicitly defining outlier measures and enhances the effectiveness of model space exploration for uncovering clusters and outliers. In addition to the discussion on RTKM, in his dissertation he describes other methods such as STKM or Spatio-Temporal K-Means as other examples of robust methods.

Another research paper from 2019 mentions that Trimmed C-Means and Trimmed K-Means implement LTS criteria within them. This journal entitled “A unified approach for cluster-wise and general noise rejection approaches for k-means clustering” implements LTS only conceptually in its trimming, not incorporating the LTS method as a pre-processing step, as seen in the previous journal [6]. Similar to how Trim C-means incorporates the LTS criterion to mitigate the impact of noisy objects, one can diminish the influence of noise by excluding objects that are far away from any cluster. Ikotun, in a journal utilizing the Systematic Literature Review (SLR) method summarizing K-Means Clustering-related journals [8], 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 [9]. 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 [10]. Moreover, this journal employs Least Square (LS) and Least Trimmed Square (LTS) as comparative estimators. This topic turns out to be still frequently discussed if seen from 2006 until now there are still many researchers who discuss clustering models that are resistant to outliers. Trimmed K-Means is an algorithm inspired by the trimming concept in LTS and Minimum Covariance Determinant (MCD) according to Rousseuw’s journal [11] with the title ”Anomaly detection by robust statistics”. 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. In Figure 1, we present the research framework, offering a clear visual guide to the theoretical and methodological aspects of our study.

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 [12]. The objective of this algorithm is to minimize the average Euclidean distance of each sample from the cluster center (centroid) [13], where Euclidean distance is used when assigning a data point to a cluster, considering the distance between the i-th data point (xi) and the i-th cluster center (ci).

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

By using the K-Means Clustering algorithm, various information can be obtained, such as web keyword sources [14], image segmentation [15], customer segmentation identification in a company [16], and much more. However,

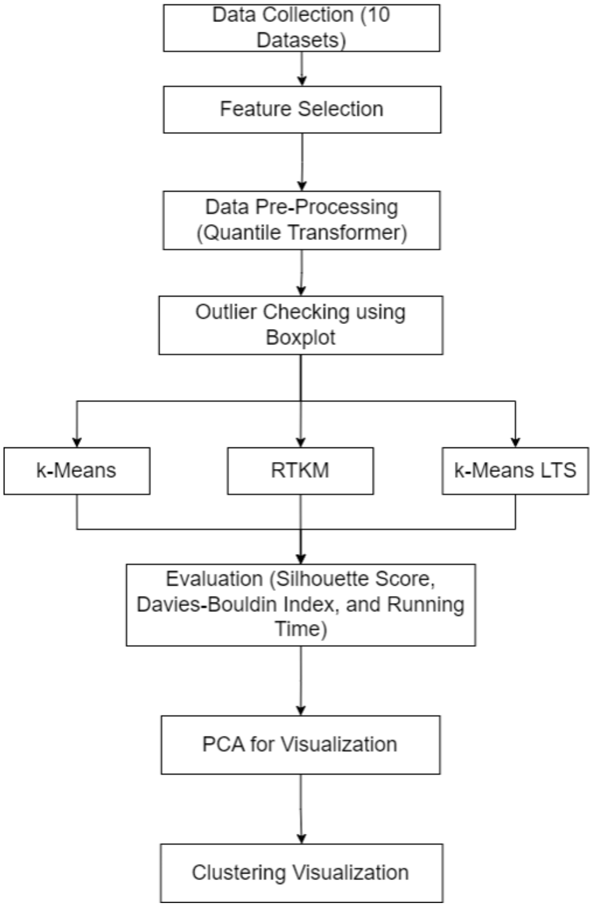
Data Collection (10 Datasets)
Feature Selection
Data Pre-Processing (Quantile Transformer)
Outlier Checking using Boxplot
k-Means
RTKM
k-Means LTS
Evaluation (Silhouette Score, Davies-Bouldin Index, and Running Time)
PCA for Visualization
Clustering Visualization

Figure 1. Research Framework

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 Rousseew in robust regression modeling susceptible to outliers as an estimator for linear coefficient [17]. 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.

$$Minimize \sum_{i=1}^{h} (r^2)_{i:n} \qquad (2)$$
$$Where (r^2)_{1:n} \leq \ldots \leq (r^2)_{n:n}$$

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 [18]. In this algorithm, determining clusters involves not only least squares (LS) but also cutting unnecessary values (trimming approach) [5]. 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 [19], 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.” In the construction of the RTKM model, there is no unsupervised determination of the optimal K value and outlier trimming percentage, and this process will be performed in the proposed model, K-Means LTS.

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 (n_percent). This means that if the data belongs to the top n_percent of the furthest distance from the centroid in the cluster, it will be trimmed from the cluster result. The search for the optimal trimming percentage will be done in an unsupervised manner, eliminating the need for multiple trials [4]. However, it will be needed to set the maximum threshold of the trimming percentage on the experiment (maximum_trimmed_percent). The algorithm proposed for determining the optimal trimming percentage is based on the largest evaluation metric result, namely the silhouette score, calculated from the each cluster of trimmed data (inliers_data). This algorithm will save the best silhouette score and best trimming percentage to be analyzed and presented in Section 5. For the algorithm's pseudocode proposed for this method, please refer to Table I.

The algorithm's complexity is manageable, as it follows a straightforward procedure outlined in Table I. By utilizing quicksort for distance calculation and simple arithmetic operations for outlier trimming, the implementation process is relatively straightforward. First, during initialization, centroids are set and cluster IDs are assigned, with a complexity of O(n * k), where n represents the number of data points and k is the number of clusters. Next, the algorithm calculates distances between data points and centroids, typically using Euclidean distances, with a complexity of O(n * k * d), where d signifies the number of dimensions. Subsequently, outlier trimming entails sorting distances and removing outliers at an optimal percentage, involving sorting complexities of O(n * log(n)) and additional iterations with a complexity of O(n). Finally, cluster assignment, akin to initialization, operates at O(n * k). This overall complexity makes K-Means LTS suitable for datasets of moderate size and dimensionality, as it efficiently handles distance calculations and outlier removal, contributing to its computational efficiency.

In general, the proposed algorithm strikes a balance between computational efficiency and effectiveness, making it suitable for various clustering tasks. Despite its iterative nature, the algorithm's computational efficiency remains commendable, particularly when compared to alternative methods that may require more complex calculations or iterations. K-Means LTS algorithm presents a promising approach for clustering tasks, offering a balance between algorithmic complexity, ease of implementation, and computational efficiency.

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 [20]. The Silhouette function is expressed as:

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

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. By looking at these two variables, the Silhouette value can provide a clear picture or result of the quality of cluster formation, where positive values indicate good placement of objects in the cluster and negative values signify sub-optimal placement.

2) *Davies-Bouldin Index:* 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).

$$DB(C) = \frac{1}{k} \sum_{i=1}^{k} max_{(i \neq j)} \frac{\Delta(C_i + C_j)}{\delta(C_i, C_j)} \tag{4}$$

Formula (4) $\Delta(C_i)$ represents the distance within each cluster, while $\delta(C_i, C_j)$ denotes the distance between clusters. Specifically, in the context of the observed Intrinsic Dimensionality Space (IDS), the centroid diameter serves as the measure for $\Delta(C_i)$, capturing the internal spread within clusters [21]. 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 [22].

TABLE I. K-Means Least Trimmed Squares Algorithm

Algorithm 1: K-Means LTS

```
function K_Means_LTS(data, n_cluster, n_percent):
    centroids, cluster_id = initialize_kmeans(data, n_cluster)
    distances = calculate_distances(data, centroids)
    sorted_data_descending = quicksort_distance(distances)
    n_rows_to_trim = int(n_percent / 100 * len(sorted_data_descending))
    inliers_data = sorted_data_descending.iloc[n_rows_to_trim:]
    return inliers_data, cluster_id

function best_K_Means_LTS(data, n_cluster, maximum_trimmed_percent):
     best_K_Means_LTS = null
     best_silhouette_score = 0
     for n_percent in range(1, maximum_trimmed_percent):
          inliers_data, cluster_id = K_Means_LTS(data, n_cluster, n_percent)
          silhouette_score = calculate_silhouette(inliers_data, cluster_id)
          if silhouette_score greater than best_silhouette_scores:
               best_K_Means_LTS = inliers_data
               best_silhouette_score = silhouette_score
     return best_K_Means_LTS, best_silhouette_score
```

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 [23]. 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 [24]. 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). It should be emphasized that visualizations might not fully capture every aspect of clustering performance, especially in multidimensional contexts. Thus, employing supplementary evaluation metrics and methodologies beyond visualization alone is advised to ensure a thorough analysis.

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 II 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	Features	No. Data	Source
1	Air Company Customer Info	3	62988	Kaggle [25]
2	NSE Banking Sector	15	41231	NSE [26]
3	CC General	18	8950	Kaggle [27]
4	Sonar	61	208	UCI ML [28]
5	Wholesale Customers	8	440	UCI ML [29]
6	Obesity Data	17	2111	Survey [30]
7	Diabetes	9	768	NIDDK [31]
8	Nurse Stress Prediction Wearable Sensors	6	2000	Empatica E4 [32]
9	Marketing Campaign	29	2240	iFood Brain [33]
10	Bank Customer Segmentation	9	5304	Kaggle [34]

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 [35]. 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% as the maximum_trimmed_percent from Table I. with an increment value of five percent for each iteration. Even though the optimal percentage values were carried out unsupervised by comparing the silhouette scores, we observed that the optimal trimmed percentages for both RTKM and K-Means LTS algorithms consistently yielded the highest silhouette scores at the maximum percentage 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, Davies-Bouldin Index, and cluster visualizations, providing insights into how these methods respond to outliers in the clustering data.

The evaluation metrics for the K-Means, RTKM, and K-Means LTS algorithms for the ten datasets used are depicted in Tables III, IV, V, respectively. These tables present the overall results of the experiments, including the outcomes of the best number of clusters (K) and the optimal trimmed percentage (%), as well as the silhouette score and Davies-Bouldin Indices (listed under the column DBI) obtained in the experiment. The running time of finding the optimal trimmed percentage on two novel algorithms, RTKM and K-Means LTS, is also measured to examine the efficiency of the model on Table IV and V.

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. The highest silhouette value obtained by K-Means is 0.69 on the first dataset. Meanwhile, RTKM gets 0.75 for the highest Silhouette value. And K-Means LTS 0.78 on the same dataset. This observation suggests that the K-Means LTS algorithm operates optimally, leading to the formation of well-defined and distinct clusters. The value generated by k-Means LTS is close to 1.0 and is positive. Thus, the results obtained can be called optimal. 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.

The interpretation of results may be influenced by the sensitivity of the silhouette score to variations in cluster shapes and densities, which may not consistently reflect the underlying distribution of the data. The silhouette score in this experiment is also used as a reference for finding the optimal trimming percentage for the proposed K-Means LTS algorithm. The trimming percentage value is influenced by the amount of data trimmed as outliers; as more data is trimmed, the resulting clusters from inliers become denser, leading to higher silhouette scores. Hence the maximum trimming percentage of 30% used in iterations on each dataset consistently chosen as the optimal trimming percentage. This suggests that there is room for further improvement in determining a more equitable trimming percentage beyond relying solely on the silhouette score metric.

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. Unlike the silhouette value, the Davies-Bouldin Index evaluates model performance by looking at which model produces the smallest value. The smaller the value obtained by the model, the better the clustering quality. K-Means produces the smallest Davies-Bouldin Index on the first dataset, which is 0.4076. RTKM produces the smallest value on the same dataset, which is 0.3572. While K-Means LTS on the same dataset, produces the smallest Davies-Bouldin value of other modelling which reaches 0.2839. 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.

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: Air Company Customer Info, NSE Banking Sector, and Bank Customer Segmentation. We reduced the features of these three datasets to two dimensions via PCA for ease of visualization. However, it's important to acknowledge the limitations of visualizations in capturing all aspects of clustering performance, especially in multidimensional spaces even when using PCA. While scatterplots offer valuable insights into cluster distributions and separations, they may not fully represent the intricacies of clustering algorithms' behavior, especially in high-dimensional spaces. Therefore, a comprehensive analysis of clustering performance should also consider other evaluation metrics and techniques beyond visualization alone.

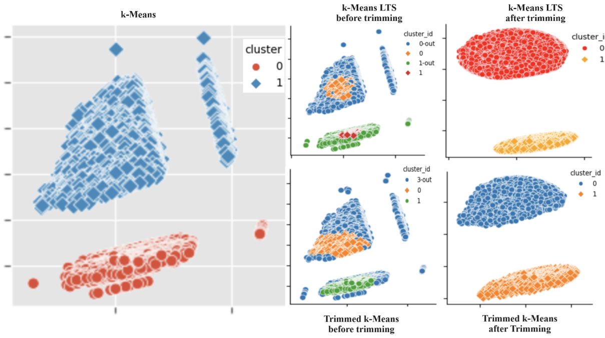

Figure 2. Visualization of K-Means, K-Means LTS, and RTKM results on the first dataset.

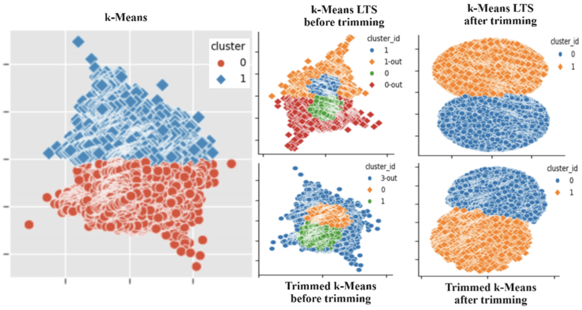

Figure 3. Visualization of K-Means, K-Means LTS, and RTKM results on the second dataset.

Based on these visualizations, K-Means clusters the data still with outliers that cannot be removed. RTKM is visually good because there are no more outliers in each group of data formed. 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

TABLE III. EVALUATION METRICS FOR K-Means

No	Dataset 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	Dataset Name	Optimal %	Silhouette	DBI	Runtime
1	Air Company Customer Info	30	0.7516	0.3572	617.5s
2	NSE Banking Sector	20	0.3965	0.9639	257s
3	CC General	30	0.3866	1.0826	26.9s
4	Sonar	15	0.1223	2.4719	0.3s
5	Wholesale Customers	25	0.7108	0.4354	0.3s
6	Obesity Data	25	0.3923	1.1222	2.0s
7	Diabetes	25	0.5664	0.6857	1s
8	Nurse Stress Prediction Wearable Sensors	25	0.4458	0.7168	7.7s
9	Marketing Campaign	15	0.5962	1.002	2.3s
10	Bank Customer Segmentation	30	0.6416	0.5074	27.4s

TABLE V. EVALUATION METRICS FOR K-Means LTS

No	Name	Optimal %	Silhouette	DBI	Runtime
1	Air Company Customer Info	30	0.7844	0.2839	188.8s
2	NSE Banking Sector	30	0.4253	0.8902	98s
3	CC General	25	0.4121	1.0762	6.4s
4	Sonar	30	0.1275	2.1569	0.4s
5	Wholesale Customers	30	0.7250	0.4735	0.3s
6	Obesity Data	30	0.3811	1.3621	0.7s
7	Diabetes	30	0.5474	0.7305	0.3s
8	Nurse Stress Prediction Wearable Sensors	30	0.5634	0.5730	0.6s
9	Marketing Campaign	5	0.5096	1.3622	0.6s
10	Bank Customer Segmentation	30	0.6672	0.4538	2.6s

clusters according to the specified parameters. For example, using K = 3 for the first dataset ”Air Company Customer Info” 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.

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

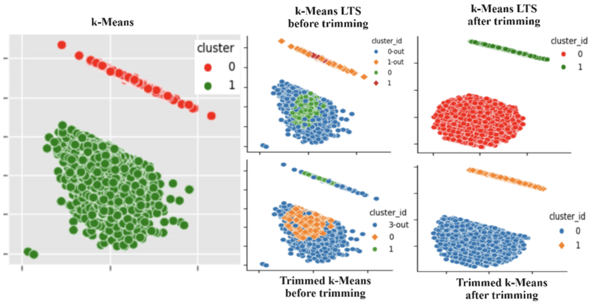

Figure 4. Visualization of K-Means, K-Means LTS, and RTKM results on the tenth dataset.

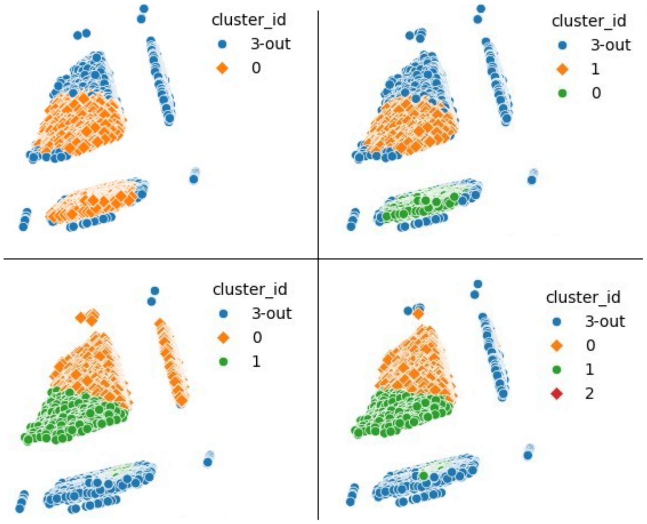

Figure 5. RTKM generated with K = 3 on the first dataset shows 1, 2, and 3 centroids with inconsistent cluster result.

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. K-Means gives unsatisfactory results and there are still many outliers that cannot be resolved despite using quantile transformers to change the data distribution to normal. It is evident that regular K-Means is not enough to resolve the outliers. RTKM works quite well although the cluster division results are not clear for some truncation percentage trials. It also cannot automatically determine the K value and truncation percentage. This K-Means LTS algorithm proposed can work better in this research through proving Silhouette Score, DBI, runtime (compared to RTKM) and also consistent visualization/cluster creation. 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 also 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 from Olga Dorabiala in [19] 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. Furthermore, a deeper examination of the outcomes generated by the proposed K-Means LTS algorithm in this study could involve exploring additional datasets with diverse sizes and characteristics. This exploration would provide insights into the algorithm's suitability for real-world applications. Moreover, it would facilitate an assessment of how the algorithm's architecture could be modified to support different types of health data and exchange patterns.

Code Availability

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

Acknowledgement

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] A. Smiti, "A critical overview of outlier detection methods," *Computer Science Review*, vol. 38, p. 100306, 2020.

[2] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020.

[3] P. S. Prateek R. Srivastava and G. A. Hanasusanto, "A robust spectral clustering algorithm for sub-gaussian mixture models with outliers," *Operation Research*, vol. 71, no. 1, p. 224, 2023.

[4] J. Kim, R. Krishnapuram, and R. Davé, "Application of the least trimmed squares technique to prototype-based clustering," *Pattern Recognition Letters*, vol. 17, no. 6, pp. 633–641, 1996.

[5] A. Banerjee and R. N. Davé, "The feasible solution algorithm for fuzzy least trimmed squares clustering," in *IEEE Annual Meeting of the Fuzzy Information, 2004. Processing NAFIPS'04.*, vol. 1. IEEE, June 2004, pp. 222–227.

[6] S. Ubukata, "A unified approach for cluster-wise and general noise rejection approaches for k-means clustering," *PeerJ Computer Science*, vol. 5, p. e238, 2019.

[7] O. Dorabiala, "Robust approaches for unsupervised learning," Ph.D. dissertation, University of Washington, 2023.

[8] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, vol. 622, pp. 178–210, 2023.

[9] N. H. Shrifan, M. F. Akbar, and N. A. M. Isa, "An adaptive outlier removal aided k-means clustering algorithm," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6365–6376, 2022.

[10] N. F. M. Azmi, H. Midi, and N. F. Ismail, "The performance of clustering approach with robust mm-estimator for multiple outlier detection in linear regression," *Jurnal Teknologi*, vol. 45, no. C, pp. 15–28, December 2006.

[11] P. J. Rousseeuw and M. Hubert, "Anomaly detection by robust statistics," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 2, p. e1236, 2018.

[12] T. S. Madhulatha, "Comparison between k-means and k-medoids clustering algorithms," in *International Conference on Advances in Computing and Information Technology*. Berlin, Heidelberg: Springer Berlin Heidelberg, July 2011, pp. 472–481.

[13] S. Misra, H. Li, and J. He, *Machine learning for subsurface characterization*. Gulf Professional Publishing, 2019.

[14] L. Li, J. Wang, and X. Li, "Efficiency analysis of machine learning intelligent investment based on k-means algorithm," *Ieee Access*, vol. 8, pp. 147 463–147 470, 2020.

[15] R. C. Hrosik, E. Tuba, E. Dolicanin, R. Jovanovic, and M. Tuba, "Brain image segmentation based on firefly algorithm combined with k-means clustering," *Stud. Inform. Control*, vol. 28, no. 2, pp. 167–176, 2019.

[16] Y. Li, X. Chu, D. Tian, J. Feng, and W. Mu, "Customer segmentation using k-means clustering and the adaptive particle swarm optimization algorithm," *Applied Soft Computing*, vol. 113, p. 107924, 2021.

[17] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005.

[18] J. A. Cuesta-Albertos, A. Gordaliza, and C. Matrán, "Trimmed *k*-means: an attempt to robustify quantizers," *The Annals of Statistics*, vol. 25, no. 2, pp. 553–576, 1997.

[19] O. Dorabiala, J. N. Kutz, and A. Y. Aravkin, "Robust trimmed k-means," *Pattern Recognition Letters*, vol. 161, pp. 9–16, 2022.

[20] G. Ogbuabor and F. N. Ugwoke, "Clustering algorithm for a healthcare dataset using silhouette score value," *Int. J. Comput. Sci. Inf. Technol*, vol. 10, no. 2, pp. 27–37, 2018.

[21] S. Petrovi´c, "A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters," in *Proceedings of the 11th Nordic workshop of secure IT systems*, vol. 2006. Citeseer, 2006, p. 53.

[22] M. Mughnyanti, S. Efendi, and M. Zarlis, "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. IOP Publishing, 2020, p. 012128.

[23] P. Bholowalia and A. Kumar, "Ebk-means: A clustering technique based on elbow method and k-means in wsn," *International Journal of Computer Applications*, vol. 105, no. 9, 2014.

[24] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, 1993.

[25] M. Aprialdi, "Customer segmentation," 2020. [Online]. Available: https://kaggle.com/competitions/sa-customer-segmentation

[26] S. Dey, "National stock exchange dataset - banking sectors," 2021. [Online]. Available: https://www.kaggle.com/datasets/sumandey/national-stock-exchange-banking-sectors/data

[27] A. Bhasin, "Credit card dataset for clustering," 2018. [Online]. Available: https://www.kaggle.com/datasets/arjunbhasin2013/ccdata/data

[28] T. Sejnowski and R. Gorman, "Connectionist Bench (Sonar, Mines vs. Rocks)," UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C5T01Q.

[29] M. Cardoso, "Wholesale customers," UCI Machine Learning Repository, 2014, DOI: https://doi.org/10.24432/C5030X.

[30] F. M. Palechor and A. de la Hoz Manotas, "Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from colombia, peru and mexico," *Data in Brief*, vol. 25, p. 104344, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340919306985

[31] M. Das, G. Bhattacharyya, R. Gong, R. Misra, S. K. Medda, S. Banik, and R. N. Das, "Determinants of gestational diabetes pedigree function for pima indian females," *Internal Medicine – Open Journal*, vol. 6, no. 1, p. 9–13, Dec. 2022. [Online]. Available: http://dx.doi.org/10.17140/IMOJ-6-121

[32] S. Hosseini, R. Gottumukkala, S. Katragadda, R. T. Bhupatiraju, Z. Ashkar, C. W. Borst, and K. Cochran, "A multimodal sensor dataset for continuous stress detection of nurses in a hospital," *Scientific Data*, vol. 9, no. 1, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1038/s41597-022-01361-y

[33] N. Boaz, “ifood data business analyst test,” 2020. [Online]. Available: https://github.com/nailson/ifood-data-business-analyst-test/tree/master

[34] S. Bansal, “Bank customer segmentation,” 2022. [Online]. Available: https://www.kaggle.com/datasets/shivamb/bank-customer-segmentation

[35] K. Bogner, F. Pappenberger, and H. L. Cloke, “The normal quantile transformation and its application in a flood forecasting system,” *Hydrology and Earth System Sciences*, vol. 16, no. 4, pp. 1085–1094, 2012.

Tricia Estella urrently 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 Ghayatrie 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 is a Member 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 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 in Bina Nusantara University. He remains actively engaged in research, focusing on machine learning, optimization, operations research, multivariate data analysis, data mining, computational intelligence, and artificial intelligence.