Enhancing K-means Clustering Results with Gradient Boosting: A Post-Processing Approach

Mousa Alzakan, Hissah Almousa*, Arwa Almarzoqi, Mohammed Alghasham,
Munirah Aldawsari, Mohammed Al-Hagery
Department of Computer Science-College of Computer, Qassim University,
Buraydah 51452, Saudi Arabia

Abstract—**As the volume and complexity of data continue to grow exponentially, finding efficient and accurate clustering algorithms has become crucial for many applications. K-means clustering is a widely used unsupervised machine learning technique for data analysis and pattern recognition. Despite its popularity, k-means suffers from certain limitations, such as sensitivity to initial conditions, difficulty in determining the optimal number of clusters, and the potential for misclassification. This research paper proposes an enhanced approach for improving the accuracy and performance of the k-means clustering algorithm by incorporating post-processing techniques using a gradient boosting algorithm. The proposed method comprises training the gradient boosting model on the labeled training set, i.e., the samples with correct cluster assignments obtained from the k-means algorithm, to predict the correct cluster assignments for the misclassified samples in the testing set. This results in refined cluster assignments for the testing set. The k-means algorithm is only used initially to cluster the data and obtain initial cluster assignments. The effectiveness of the proposed approach is validated through experiments on several benchmark datasets, and the results show a significant improvement in clustering accuracy and robustness compared to the standard k-means algorithm. The proposed approach has the potential to enhance the performance of k-means in various real-world applications and domains.**

Keywords—K-means; gradient boosting; post-processing; misclassification; machine learning

I. Introduction

Enhancing the performance of clustering algorithms has become essential for obtaining accurate and effective clustering in the era of contemporary data, with its growing amount and diversity. Clustering is a strategy that is used for analyzing data, collecting similar data points together, and recognizing patterns in data that would otherwise be invisible [1]. There are several types of clustering algorithms, such as hierarchical methods, density-based methods, grid-based methods, and partitioning-based methods, each of which differs in the way they measure the similarity or distance between entities [2].

The k-means algorithm is an unsupervised algorithm that was improved by MacQueen in 1967 [3]. The k-means algorithm is a type of partitioning-based method that is used to group similar data. Each group of data is called a cluster. In the first stage, the algorithm randomly assigns an initial set of points for k clusters based on the nearest center of the clusters. Then, it modifies the points until it reaches the nearest cluster center. The process is iterative and continues until the centroids no longer update, resulting in the final centroids representing the ultimate centers of k clusters. The function of updating and modifying centroids is performed by calculating the distance measure specified in the algorithm [4].

The k-means algorithm offers several advantages, including its speed, simplicity, and ease of implementation [5]. It is useful in different applications such as marketing, recommendation systems, smart city services, the analysis of business data, and the analysis of user behaviors [6]. However, even with the advantages of the k-means algorithm, it still faces some drawbacks, such as its problem with local optima and its sensitivity to initial centroids. The k-means algorithm is sensitive to the centroids and will give different results whenever the initial centroids change [5]. In this study, we build upon a previous research paper that explored techniques for enhancing the k-means algorithm [7]. While acknowledging the contributions of the original work, we present an extended methodology that incorporates optimization techniques to further improve the clustering outcomes. The aim of this study is to surpass the performance achieved in the earlier study.

Several previous studies have been conducted in the field of improving the k-means algorithm. In [7], the authors presented a novel concept of post-processing the clusters obtained by the classical k-means algorithm to improve the quality of the resulting clusters. The post-processing approach consists of four steps that combine the k-means algorithm with a supervised learning algorithm, resulting in hybrid k-means-supervised learning (KM-SML). The paper proposed an approach to extract the majority of misclassified records from the clustered dataset and post-process them using the supervised machine learning algorithm. The results obtained from applying the proposed approach demonstrated significant improvements compared to the classical k-means algorithm. Precision and recall, two evaluation metrics, were used to assess the enhancements brought by the KM-SML approach. In both cases, better results were achieved using the KM-SML approach compared to the classical k-means algorithm. In [8], the researchers suggest a method that combines an optimization algorithm, namely Particle Swarm Optimization (PSO), with a k-means clustering algorithm. According to the comparison analysis, using PSO to determine the initial centroids yields promising results. While other studies highlight the benefits of combining metaheuristic optimization algorithms and data mining techniques, opening avenues for further research in this field, in [9], the researchers propose the integration of nature-inspired optimization algorithms, such as ant, bat, cuckoo, firefly, and wolf search algorithms with k-means clustering to overcome the drawback of getting stuck at local optima determined by random initial centroids. By combining these algorithms with k-means, the researchers aim to achieve unprecedented performance enhancements in terms of clustering accuracy. The results of the evaluation experiments show significant improvements in performance, particularly for the C-Bat and C-Cuckoo hybrid algorithms.

It is evident from the literature that researchers integrate various techniques with the k-means algorithm to achieve better outcomes or performance; Likewise, based on the previous work [7], this paper presents an enhanced approach for increasing the accuracy of the k-means algorithm by utilizing post-processing techniques with the gradient boosting algorithm. The proposed approach is implemented using the Python programming language and the Scikit-learn library [10] and applied to three datasets, which are the Iris dataset, Forest, and Banknote datasets [11], [12], and [13] respectively. The proposed approach implements the Split Criterion (SC) [7] for detecting potentially misclassified points. Additionally, the paper utilizes an extra set of threshold values for the SC and compares the results to other approaches [7], [14].

The proposed approach calculates the Euclidean distance of data points and the centroids of k clusters and scales the data using MinMaxScaler. Based on the SC threshold, the k-means results from the datasets are separated into training and testing sets. If the value of SC exceeds a predetermined threshold, the data point is considered a misclassified point and transferred to the test set, while the correct labels are transferred to the train set. The labels generated are used to train the gradient boosting classifier. As a result, the approach reached up to 97% accuracy on the Iris dataset. The approach presented in this paper assists in improving the performance of k-means clustering by minimizing the number of misclassified points, which helps to increase the accuracy of the algorithm.

The rest of the paper is organized as follows: Section II presents the literature review. Section III describes the techniques used in this paper. Section IV presents the results and discussion. Section V covers the conclusions.

II. Literature Review

The k-means method is an unsupervised clustering algorithm. It is extensively used in data mining because it is easy to use and understand and due to its applicability to various application domains [15]. The k in k-means represents the number of resulting clusters. The k-means algorithm accepts unlabeled data and groups it into k non-overlapping groups called clusters based on how close each point in a cluster is to the mean, called the centroid center, of that cluster [16].

In numerous papers, the k-means algorithm is combined with another algorithm to enhance execution efficiency, improve results, or achieve both. In [17], k-means and long short-term memory (LSTM) neural networks are used to analyze the behavior of electricity consumption for generating targeted marketing and recommending usage strategies. The data is first clustered using the k-means algorithm. Then, it is labeled based on a previous dataset and fed into LSTM to produce the results. The results are more accurate and efficient than using the LSTM directly. Instead of using LSTM, [18] uses a hybrid method that employs k-means with the Gaussian mixture model (GMM) for detecting malignant and benign breast cancer tumors using mammographic images. This approach has higher accuracy, signal-to-noise ratio, and a lower error rate than non-hybrid existing techniques such as k-means, GMM, and thresholding.

While [18] uses a hybrid model of k-means and GMM, [19] employs a hybrid model based on two evolutionary algorithms. It uses the fireworks-based and cuckoo-search-based evolutionary algorithms to improve the quality of the resulting clusters. In addition to these two algorithms, the method in [19] selects representatives of data using instance reduction to solve the empty cluster issue. The empty clusters problem happens when the number of clusters increases [20]. Moreover, this method enhances the selection of the initial centroids by using heuristics alongside evolutionary-based algorithms.

Both [21] and [22] use the Support Vector Machine (SVM) method with the k-means algorithm. Both techniques use k-means to cluster the values before inputting them into the SVM algorithm. In [21], the approach is to monitor and predict student performance in higher education. The resulting clusters from the k-means algorithm are further analyzed using SVM to accurately classify students as high-performing or low-performing students, which produces more accurate results than using the SVM only, whereas [22] uses the k-means algorithm on unlabeled data to generate a subset of the significant features to be the training set for the SVM instead of the complete dataset. According to [22], this approach improves the classification accuracy and performance in some situations compared to other approaches such as C-SVC and S4VM.

Unlike [21] and [22], which use SVM after k-means, [7] utilizes a supervised learning technique, in particular, the random forest classifier [23] is employed to improve the results of k-means. In addition, [7] proposes a method to detect potentially misclassified points. After applying the k-means algorithm to the Iris dataset [11], the results are examined for any potentially misclassified points. The detection of the misclassified points is done as follows, for each of the chosen minimum distances, divide each by the minimum distance to each cluster. If the values cross a predetermined threshold, then there is a possibility of misclassification for this point. After determining the possible misclassified points, they are extracted from the dataset, and the supervised learning algorithm is trained with the correctly clustered data. Finally, the misclassified points are entered into the model for classification. This proposed approach produces more accurate results than using the k-means method exclusively.

The k-means algorithm is sensitive to the initial clustering centers since the initial selection of centroids can affect the number of iterations and execution time [16]. To reduce the number of iterations and the running time, [6] have proposed reducing the dimensions of the data using percentile techniques and the Principal Component Analysis (PCA). The centroids are selected from the resulting reduced data. This technique has better results than both random and k-means++ initializations.

Another issue related to the selection of the cluster centers is that the non-optimal choice of centers leads the algorithm to converge to local minima [16]. Therefore, it is imperative to select the optimal centroid location to avoid getting stuck in local minima. The author in [14] proposes a method to determine optimal centers. This method employs an ant colony algorithm and uses positive and negative pheromone feedback to optimize the initialization of centroids. An additional issue is the instability of the assignment of clusters [16]. To overcome instability, [24] combines density and multiple clustering. This solution improves the running time and stability of the clustering by choosing the centroids according to the furthest distance and the highest density principle. However, solutions that use just density have a high time complexity [24].

Determining a suitable number of clusters requires domain knowledge [25]. Unfortunately, domain knowledge is not always readily available. To mitigate this issue, [26] proposes a method that does not require the manual specification of the number of clusters. One notable benefit of employing this method is its ability to accelerate the execution process and improves accuracy. It outperforms k-means when the data has lower dimensionality. Another proposed approach that does not require the specification of the number of clusters is in [27]. In [27], the authors propose and test a fully unsupervised k-means algorithm that does not need initialization and parameter selection. It auto-determines the optimal number of clusters using the entropy concept. In addition, it has good results when compared with the existing methods.

Traditional k-means implantations use the Euclidean distance to find the distances between the points [28]. However, [29] opted to use the evidence distance, which can deal with uncertainty. Instead of using the Euclidean distance, the method utilizes the evidence distance, resulting in higher accuracy and a reduced number of iterations. In contrast, [30] have proposed a k-means algorithm, L2-weighted k-means, whose mean is computed using the weighted feature space transformation. The L2-weighted k-means algorithm described in [30] was used to help in drilling for groundwater. Specifically, it was used to find the capacity of the average digging per day and to optimize profitability and productivity.

The authors of [31] state that the Lloyd algorithm for k-means does not perform well in dealing with large data. Therefore, [31] presents a k-means algorithm that uses neighbor information for assigning and updating the points. This algorithm reduces the distance calculations and increases the accuracy of the produced neighbors.

MapReduce, a programming model for parallel and distributed clusters, and Hadoop, a framework for distributed processing and storage of big data [32], have been used to enhance the scalability and parallelize the execution of k-means methods, as demonstrated in several studies [5], [19], [33]. The author in [5] describes a technique for news classification that uses MapReduce and Hadoop for parallelization. It also improves the selection of the initial centroids by leveraging the organizational structure of the data. The results show a 30% decrease in execution time over the method that does not employ parallelism.

Despite these efforts to enhance the performance of the k-means algorithm, gaps persist in the existing literature, including the reliance on domain knowledge for parameter selection or initialization, which limits applicability across diverse domains. Additionally, few methods provide a unified solution to address multiple shortcomings of k-means, such as sensitivity to initial conditions and misclassification issues. The proposed approach aims to bridge these gaps by introducing a post-processing technique using gradient boosting to refine cluster assignments obtained from k-means. Unlike prior methods that focus on specific enhancements or manual parameter tuning, the approach offers a comprehensive solution to improve clustering accuracy and robustness across various datasets and application domains.

III. Methodology

By adopting the data analysis techniques and the clustering approach, this research paper proposes an improvement to the performance of the k-means clustering algorithm by using gradient boosting in post-processing. The proposed approach intends to improve the quality of the k-means by post-processing the resulting clusters, which will contribute to delivering new insights in the context of clustering problems. This section describes the overall methodological approach of the present research paper by covering six fundamental elements. Section III-A describes the utilized datasets. The k-means clustering algorithm is then described in Section III-B. Section III-C provides an in-depth explanation of the split criterion technique. As well, Section III-D illustrates the post-processing methodology. Section III-E presents the evaluation matrices used to assess the proposed model. Eventually, the experimental setup is presented in Section III-F.

A. Datasets

The proposed approach is examined by using three benchmark datasets from the UCI Machine Learning Repository, which are popular datasets in the machine learning community. Namely, the Iris, Forest, and Banknote datasets [11], [12], and [13]. Table I describes the characteristics of each dataset, including the number of instances, the number of attributes, and the number of clusters for k-means. Additionally, a normalization technique is applied to the datasets in order to facilitate and improve the classification. The normalization is accomplished by using MinMaxScaler from the Scikit-learn toolkit [10] to scale each feature between 0 and 1. Consequently, the k-means outcomes utilizing the datasets are classified into training and testing sets based on SC results and the selected SC threshold. If the SC result of any point is higher than the predetermined SC threshold, the point will be added to the misclassified points, which are defined as the test sets to be used in the process of testing in the post-processing phase, while the correct labels are defined as the training set in the training process in the post-processing.

TABLE I. Datasets Description

Dataset	Instances	Attributes	Initial k
Iris [11]	150	4	k = 3
Forest [12]	523	27	k = 4
Banknote [13]	1372	5	k = 2

B. K-means Clustering Algorithm

The k-means algorithm is perhaps the most widely utilized clustering method. It has been explored for several decades. Therefore, it serves as the basis for several advanced clustering techniques [34]. The k-means algorithm is widely used because it uses straightforward, non-statistical principles, is extremely adaptable and flexible, and performs well. Furthermore, [34] mentions that the k-means algorithm is essentially composed of two phases. First, it assigns points to an initial set of k clusters. Second, it modifies and updates the assignment points. The process of assigning points is based on the nearest cluster center according to the distance function. Traditionally, k-means clustering uses Euclidean distance to compute the distance between points and the cluster centers [34]. Eq. (1) shows the distance metric formula used in this paper.

$$dist(x_i, y_c) = \sqrt{\sum_{j=1}^{a}(x_{ij} - y_{cj})^2}$$
$$i = 1, ..., n;\ c = 1, ..., k \tag{1}$$

where:

x is the data point

y is the centroid

n is the number of points

k is the number of clusters

a is the number of attributes

Consequently, updating and assigning points take place repeatedly until the cluster fitness is no longer improved by changes. The procedure ends at this stage, and the clusters are complete. Listing 1 shows the optimal values for the k-means parameters that produced the best outcomes.

Listing 1: k-means Parameters

```
KMeans(n_clusters=n, init='k-means++', n_init=10, max_iter
     =300, tol=0.0001,verbose=0, random_state=0, copy_x=True
     , algorithm='lloyd')
```

C. Split Criterion

In the proposed method, the SC [7] phase in the post-processing step plays a crucial role in determining potentially misclassified points by the k-means algorithm. This phase is significant for cluster analysis as it contributes to determining the accuracy of the clustering algorithm. It does this by finding and separating the likely misclassified points so they can be used as input for the final phase in post-processing.

Upon the completion of the k-means algorithm, k groups are generated, each of which comprises a center and a set of data points. In order to enhance the accuracy of clustering, misclassified data points must be identified and corrected. Here is where the SC method is applied.

The SC method begins by calculating the Euclidean distance between each data point and the centers of all clusters. For each point x, the minimum distance from it to each center is determined and referred to as min_{xc}. Then, min_{xc} is divided by the distance of each centroid to the point. This yields values between 0 and 1, representing the ratio of the minimum distance from point x to cluster c to each cluster center.

A threshold value between 0 and 1 is chosen to identify the misclassified data points. If any of the values calculated for data point x exceeds the chosen threshold, then x is considered misclassified. However, the point with the minimum distance to a cluster to which the point belongs is excluded from the comparison with the threshold since it will also result in 1.

For instance, if a point x_1 and three centers c_1, c_2, and c_3 are given, the distance between this point and the three cluster centers is calculated, resulting in three distances: $dist(x_1, c_1)$, $dist(x_1, c_2)$, and $dist(x_1, c_3)$. Then, the minimum distance, say $dist(x_1, c_2)$, is determined, and the SC result R of each distance can be calculated as follows:

$$SC(x_1, c_1) = \frac{dist(x_1, c_2)}{dist(x_1, c_1)} = R \quad 0 \le R \le 1 \tag{2}$$

$$SC(x_1, c_2) = \frac{dist(x_1, c_2)}{dist(x_1, c_2)} = R \qquad R = 1 \tag{3}$$

$$SC(x_1, c_3) = \frac{dist(x_1, c_2)}{dist(x_1, c_3)} = R \quad 0 \le R \le 1 \tag{4}$$

If the value obtained from Eq. (2) or (4) exceeds the specified threshold, the data point x_1 is considered misclassified. The minimum distance to the cluster, represented as $dist(x_1, c_2)$, is excluded from the comparison according to Eq. (3), resulting in a value of 1.

The SC method is a (moderate) technique for identifying misclassified data points in k-means clustering. By utilizing the threshold value, the SC technique can identify misclassified points so they can be minimized, thereby improving the accuracy of the clustering algorithm.

D. Post-processing Approach

Post-processing is a technique used with clustered data of k-means to improve the accuracy and quality of the resulting clusters [7]. In this phase, possibly misclassified labels are detected, and a corrective process is applied to obtain more accurate results. This study incorporates gradient boosting as a post-processing technique after applying the SC method. Gradient boosting is a popular machine learning method utilized for regression and classification tasks. It involves combining multiple weak models, usually decision trees, to form a powerful model that can make precise predictions. Gradient boosting is effective in handling imbalanced datasets, noisy data, and high-dimensional data [35].

To determine the optimal number of estimators for gradient boosting, a method is employed where the data is divided into training and testing sets, and multiple iterations of training and testing are conducted. Various ranges of estimators are tested, and the results are compared to identify the ideal number. The experiments indicate that the best outcomes were obtained with 100-200 estimators.

Gradient boosting also requires additional parameters such as learning rate, maximum depth, and random state. The best values for these parameters are found using a grid search technique, which entails trying a range of values for each parameter and choosing the combination that results in the highest performance [36]. However, in this study, the default values provided by the library were used for these parameters, as they are generally well-suited for a wide range of scenarios and models, as shown in Listing 2.

Listing 2: Gradient Boosting Parameters

```
ensemble.GradientBoostingClassifier(loss='log_loss',
    learning_rate=0.1, n_estimators=100, subsample=1.0,
    criterion='friedman_mse', min_samples_split=2,
    min_samples_leaf=1, min_weight_fraction_leaf=0.0,
    max_depth=3, min_impurity_decrease=0.0, init=None,
    random_state=None, max_features=None, verbose=0,
    max_leaf_nodes=None, warm_start=False,
    validation_fraction=0.1, n_iter_no_change=None, tol
    =0.0001, ccp_alpha=0.0)
```

The k-means clustering algorithm's performance was successfully enhanced by employing gradient boosting during the phase of post-processing. The proposed approach substantially increased algorithm accuracy through the detection of potentially misclassified labels and the attempts to correct them. Algorithm 1 demonstrates the process of training, testing, and evaluating the performance of the gradient boosting algorithm in the post-processing phase.

Algorithm 1 Gradient Boosting in the Post-processing Phase

```
Input: Correctly labeled set $X_{train}, y_{train}$ and the misclassified set $X_{test}$
Output: Predicted labels $y_{pred}$ for all dataset
Set the parameters for the gradient boosting algorithm
Train the gradient boosting classifier on $X_{train}$ and $y_{train}$
$y_{test} \leftarrow$ Apply the trained gradient boosting classifier on $X_{test}$
$y_{pred} \leftarrow$ APPEND($y_{train}$, $y_{test}$)
Evaluate classifier performance using evaluation metrics (accuracy, precision, recall, F1-score) on the corrected labels.
return $y_{pred}$
```

E. Evaluation Metrics

The performance of the proposed approach is evaluated in terms of classification accuracy, precision, recall, and F1 scores to the formerly indicated datasets. The evaluation metrics are calculated using the following equations, which are measured by utilizing the true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Accuracy represents the number of correctly classified data instances over the total number of data instances. Eq. (5) shows the accuracy formula:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \tag{5}$$

The precision result represents the positive predictive value in the classified data instances. Eq. (6) shows the precision formula:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

The recall value represents the true positive rate of data instances. Eq. (7) shows the recall formula:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

The F1 score represents the harmonic mean of both precision and recall. Eq. (8) shows the F1 score formula:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

The results of all the evaluation metrics used to measure the performance of the proposed approach on the previously described datasets are discussed in the Results and Discussion section.

F. Experimental Setup

For the experimental setup, the proposed method is implemented using the Python programming language. The libraries NumPy, Pandas, and Scikit-learn are chosen for their ease of use and their popularity in the machine learning community. The experiments are conducted on a computer with an Intel Core i7 processor and 16GB of RAM.

As mentioned previously, Iris, Forest, and Banknote are the three UCI datasets that were employed in the experiment. After obtaining clustered data with k-means, each data is split using SC into correctly classified points, a training set, and possibly misclassified points, a testing set. Then, the training set employed to train the model using the training set processed by the k-means algorithm. These labeled data are stable and will not be modified after the post-processing phase is performed. Furthermore, the testing set containing all misclassified labels is fed forward to the trained model, which modifies the labels to obtain correct and enhanced results.

The proposed approach is conducted using the following steps:

Step 1. Normalize the dataset using the MinMaxScaler.
Step 2. Process the normalized data by the k-means algorithm to produce a k number of clusters.
Step 3. Split the clustered data using the SC method. The correct labels are used as the training set for the gradient boosting algorithm, while the misclassified labels are stored for later use.
Step 4. Predict the labels for the misclassified data. The final result is obtained by combining the correct and predicted labels.

The entire process of the proposed method is shown in Fig. 1.

Eventually, the results of the experiments have demonstrated that the post-processing accompanied by SC and gradient boosting approaches is a powerful tool for enhancing the results of the k-means clustering algorithm. The approach offers a flexible and effective method to refine the results of the k-means algorithm, making it a valuable tool for various applications and datasets. A comprehensive and detailed presentation of the results is provided in the Results and Discussion section of the research.

IV. Results and Discussion

The outcomes of the experiment that has been successfully and effectively conducted, based on the mentioned steps earlier, will be detailed and compared to other approaches from [7], [14] in this section. The section is divided into three subsections. The first subsection is to show the SC results and understand the effect of various threshold values. In the second subsection, the enhanced accuracy of the supervised model employing the gradient boosting algorithm is presented

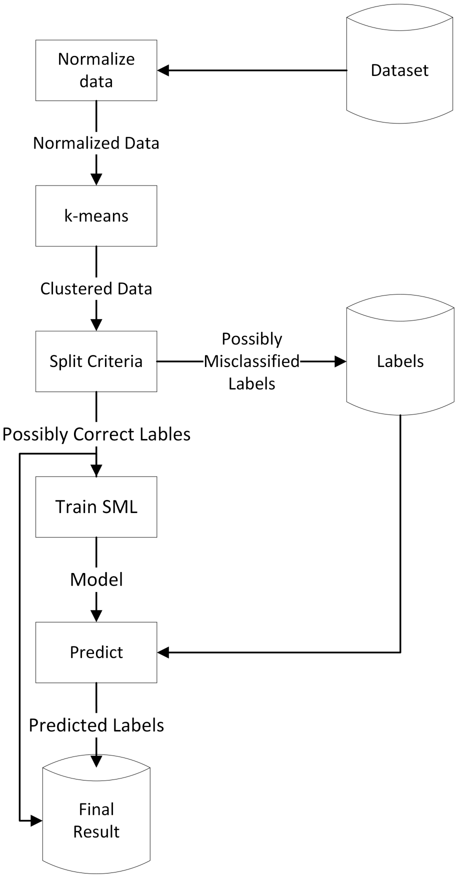


Fig. 1. A flowchart of the proposed approach.

and compared against random forest for the Iris dataset. For the last subsection, the enhanced model is compared with other improved k-means algorithms, and the model outperforms all of them in two datasets. Before showing the post-process results, the accuracy of k-means for the three datasets needs to be shown. It is as follows:

- 89% for the Iris dataset (133 out of 150 correctly clustered).
- 55% for the Banknote dataset (790 out of 1372 correctly clustered).
- 77% for the Forest dataset (405 out of 532 correctly clustered).

A. Split Criteria

The split criteria are used to detect misclassified points, but its modest and adequate mechanism could also identify correctly classified points as misclassified. Choosing a specific threshold for split criteria is not straightforward. Therefore, a range of values for the threshold is sufficient to work on all datasets. The study of split criteria has been accomplished previously for the Iris dataset using three variables: total points, correctly classified points, and misclassified points [7]. In this study, a new variable called "remaining misclassified points" is introduced, which represents the points that should have been detected as misclassified by the split criteria. Additionally, the exploration of various threshold values for the split criteria is conducted. Furthermore, the study is extended with two additional datasets, the Banknote and Forest datasets.

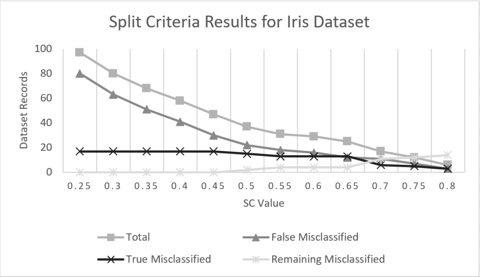


Fig. 2. Split criteria results for iris dataset.

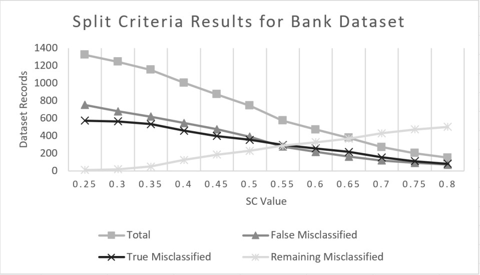


Fig. 3. Split criteria results for banknote dataset.

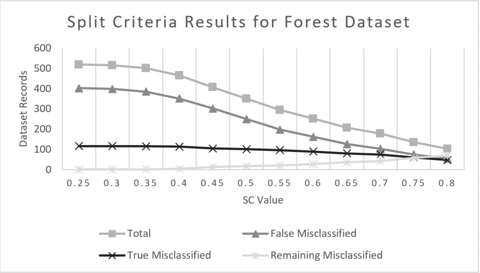
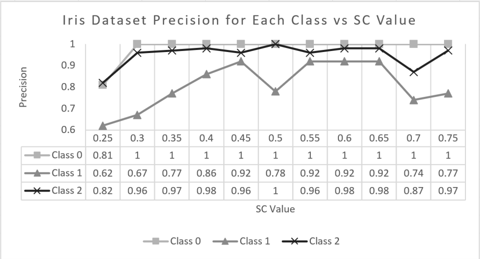


Fig. 4. Split criteria results for forest dataset.

Based on a previous study of the SC results for the Iris dataset, the appropriate threshold values for balanced results are between 0.4 and 0.6 [7]. With the previous conclusion as a guide, this study confirms that threshold values between 0.4 to 0.6 are also sufficient for all three datasets. When the threshold value is small, all misclassified points are almost detected right, true misclassified, as shown in Fig. 2, 3, and 4 precisely at 0.25 SC value. At the same time, the points that are correctly clustered by k-means are also considered as possibly misclassified points, false misclassified. Accordingly, gradually increasing the threshold value decreases both true and false misclassified while increasing the remaining misclassified points that should be detected as misclassified. Unfortunately, a compromise should be made by choosing balanced results for the threshold value with the main focus on decreasing the remaining misclassified points as much as possible, as follows: 0.45-0.65 for the Iris dataset as shown in Fig. 2, 0.35-0.50 for the Banknote dataset as shown in Fig. 3, 0.4-0.6 for Forest dataset as shown in Fig. 4. Therefore, continuing to use the same range for the threshold between 0.40 and 0.60 seems reasonable while also considering comparing results and being consistent with previous findings. In addition, the SC threshold value is incremented by 0.5. Therefore, the set of points within the two endpoints [0.40,0.60] are 0.40, 0.45, 0.50, 0.55, and 0.60.

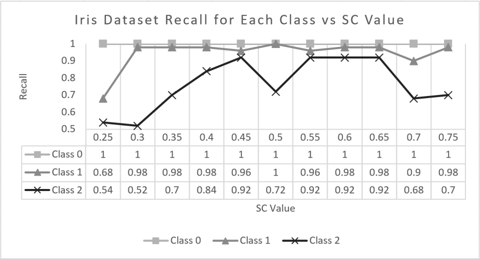
B. Gradient Boosting Post-process for the Iris Dataset

The Iris dataset has 150 data points and three classes, each with 50 data points [11]. This section focuses on presenting the results obtained from processing the Iris dataset exclusively. The results of post-process precision and recall for each class of the Iris dataset with a set of threshold points between two endpoints [0.25, 0.75] are shown in Fig. 5 and 6. The two figures represent the three classes as Class 0, Class 1, and Class 2. In addition, each figure is associated with a data table containing the precise percentage value. The data table is essential in this context as the Iris dataset has been used extensively in testing algorithms, and even a slight variation in the percentage is considered a significant accomplishment. A comparison of precision and recall results with random forest results is shown in Table II. This paper uses k-means with gradient boosting, abbreviated as K+GB, while the previous work has used k-means with random forest, abbreviated as K+RF. Besides precision and recall, the accuracy of both models is set out in Table III. Both Tables II and III illustrate the results of a set of threshold values between 0.40 and 0.60. It is apparent from both tables that a few cell values are empty. All these missing values are related to random forest results since the previous experiment did not provide the results of either 0.45 or 0.55 threshold values.



	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75
Class 0	0.81	1	1	1	1	1	1	1	1	1	1
Class 1	0.62	0.67	0.77	0.86	0.92	0.78	0.92	0.92	0.92	0.74	0.77
Class 2	0.82	0.96	0.97	0.98	0.96	1	0.96	0.98	0.98	0.87	0.97

Fig. 5. Iris dataset precision for each class vs. SC Value.



	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75
Class 0	1	1	1	1	1	1	1	1	1	1	1
Class 1	0.68	0.98	0.98	0.98	0.96	1	0.96	0.98	0.98	0.9	0.98
Class 2	0.54	0.52	0.7	0.84	0.92	0.72	0.92	0.92	0.92	0.68	0.7

Fig. 6. Iris dataset recall for each class vs. SC Value.

Class 1 and 2 misclassified points are misallocated between Class 1 and 2, while all Class 0 points hold steady along all threshold values except at 0.25, as seen in Fig. 5 and 6. What stands out in both figures is the 100% precision and recall for Class 0, except at 0.25, which has 81% precision but still has 100% recall. Obviously, misclassified points are only within Class 1 and Class 2. Class 2 has higher precision than Class 1, while Class 1 has higher recall than Class 2. Further statistics by calculating the average reveals that the model precision percentage is less than 90% at three threshold values: 0.25, 0.30, and 0.70 with 75%, 88%, and 87% precision, respectively. In contrast, the recall percentage is less than 90% at five different threshold values: 0.25, 0.30, 0.35, 0.70, and 0.75 with 74%, 83%, 89%, 86%, and 89% recall, respectively. Overall, excellent results are easily observed at four different threshold values.

Without including rows with missing values, the proposed model outperforms random forest by two out of three threshold values in Class 1 and Class 2 in Table II and accuracy in Table III. In Table II, for Class 1, the model's precision is better at 0.40 and 0.60, and its recall is better at 0.40 and 0.50. Accordingly, for Class 2, the model's precision is better at 0.40 and 0.50, and its recall is better at 0.40 and 0.60. Moreover, in Table III, its accuracy is higher for both 0.40 and 0.60. The most interesting point of these results is that the random forest's best result is at a 0.60 threshold value with 94% accuracy. In contrast, the best accuracy for the proposed model is at the same threshold value with 97% accuracy.

C. Gradient Boosting Post-process vs. Other Improved K-Means

The proposed model has been tested with two additional datasets besides the Iris dataset, and the findings are presented with the results of other approaches as benchmarks. The model's accuracy and average accuracy using a set of threshold values are reported in Table V. The average accuracy is calculated for two reasons. First, the approach implemented can not be validated by one threshold value to be viable for comparison with other approaches. Second, the other approaches used as benchmarks have presented their accuracy values by taking the average values after running the algorithms several times. Therefore, Table IV compares the obtained average accuracy from the proposed model with four other algorithms: k-means, FCM, three-way k-means, and improved three-way k-means [14]. The accuracy is the only finding reported in this section.

The proposed method using gradient boosting outperforms other algorithms with two out of three datasets for one algorithm and three out of three for the rest of the algorithms as reported in Table IV. After obtaining the accuracy for the set of points between 0.40 and 0.60 threshold values as shown in Table V, the average accuracy is calculated as 94.67% for the Iris, 63.24% for the Banknote, and 78.61% for the Forest datasets. The proposed approach outperforms all algorithms for the Iris and Banknote datasets, achieving an approximate increase in accuracy of 4% and 2%, respectively. Only for the Forest dataset, the model slightly exceeds all algorithms except for the improved three-way k-means.

V. Conclusion

Clustering algorithms are frequently used to identify dispersed patterns and group them into clusters. In order to improve the quality of the k-means clustering algorithm, this research paper has been introduced. The proposed research paper enhances the performance of the k-means clustering

TABLE II. Iris Dataset Precision and Recall for K-Means + Random Forest (K+RF) vs. K-Means + Gradient Boosting (K+GB)

EM	Precision						Recall					
Classes	Class 0		Class 1		Class 2		Class 0		Class 1		Class 2	
SC	K+RF	K+GB	K+RF	K+GB	K+RF	K+GB	K+RF	K+GB	K+RF	K+GB	K+RF	K+GB
0.40	1.00	1.00	0.84	0.86	0.95	0.98	1.00	1.00	0.96	0.98	0.82	0.84
0.45	-	1.00	-	0.92	-	0.96	-	1.00	-	0.96	-	0.92
0.50	1.00	1.00	0.86	0.78	0.98	1.00	1.00	1.00	0.98	1.00	0.84	0.72
0.55	-	1.00	-	0.92	-	0.96	-	1.00	-	0.96	-	0.92
0.60	1.00	1.00	0.85	0.92	1.00	0.98	1.00	1.00	1.00	0.98	0.82	0.92

TABLE III. Accuracy of Post-processes: Random Forest vs. Gradient Boosting

SC	K+RF	K+GB
0.40	0.93	0.94
0.45	-	0.96
0.50	0.94	0.91
0.55	-	0.96
0.60	0.94	0.97

TABLE IV. Average Accuracy Comparison between K-Means + Gradient Boosting and others

Datasets	K-Means	FCM	Three-Way k-Means	Improved Three-Way k-Means	K-Means+GBA
Iris	0.8866	0.8933	0.9040	0.9040	0.9467
Banknote	0.5758	0.5969	0.6123	0.6131	0.6324
Forest	0.7795	0.7540	0.7807	0.8294	0.7861

TABLE V. Accuracy of K-Means + Gradient Boosting for the Three Datasets

SC	Iris	Banknote	Forest
0.40	0.9400	0.6407	0.7462
0.45	0.9600	0.6465	0.8026
0.50	0.9067	0.6458	0.7838
0.55	0.9600	0.6443	0.7932
0.60	0.9667	0.5845	0.8045
Average	0.9467	0.6324	0.7861

algorithm by employing gradient boosting as a post-processing phase. Consequently, the proposed model optimizes misclassified candidate clusters from the k-means algorithm by post-processing them using the gradient boosting algorithm. Across three well-known benchmark datasets, the proposed approach performance is assessed in terms of accuracy, precision, recall, and F1 score. According to the experimental outcomes, the proposed model achieved an average accuracy of 94.67% for the Iris dataset, 63.24% for the Banknote dataset, and 78.61% for the Forest dataset. The outcomes of the proposed model confirm its effectiveness and demonstrate its applicability to a wide variety of clustering problems. Thus, several real-life domains can take advantage of the proposed model in order to enhance the data analysis process. The proposed approach has been explored on a limited number of benchmark datasets that do not encompass real-world data. For this reason, the model's capacity for generalization is likely to be optimized in future research. Eventually, based on these principles, future research will concentrate on enhancing the accuracy of the proposed model by utilizing a real-world dataset, assimilating it with other learning approaches, and offering a sophisticated split criteria technique to achieve more promising outcomes.

Acknowledgment

Researchers would like to thank the Deanship of Scientific Research, Qassim University, for funding the publication of this project.

References

[1] S. K. Majhi and S. Biswal, "Optimal cluster analysis using hybrid K-Means and Ant Lion Optimizer," *Karbala International Journal of Modern Science*, vol. 4, pp. 347–360, Dec. 2018.

[2] X. Han, L. Quan, X. Xiong, M. Almeter, J. Xiang, and Y. Lan, "A novel data clustering algorithm based on modified gravitational search algorithm," *Engineering Applications of Artificial Intelligence*, vol. 61, pp. 1–7, May 2017.

[3] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, vol. 5.1, pp. 281–298, Jan. 1967. Publisher: University of California Press.

[4] M. Lv, "Application of an K-means Improved Clustering Analysis Algorithm in the Design of Resource Management Information System," *2022 World Automation Congress (WAC), Automation Congress (WAC), 2022 World*, pp. 158–162, Oct. 2022. Publisher: TSI Enterprises.

[5] Y. Zhou, "Application of K -Means Clustering Algorithm in Energy Data Analysis," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–8, May 2022.

[6] M. Zubair, M. A. Iqbal, A. Shil, M. J. M. Chowdhury, M. A. Moni, and I. H. Sarker, "An Improved K-means Clustering Algorithm Towards an Efficient Data-Driven Modeling," *Annals of Data Science*, June 2022.

[7] I.-D. Borlea, R.-E. Precup, and A.-B. Borlea, "Improvement of K-means Cluster Quality by Post Processing Resulted Clusters," *Procedia Computer Science*, vol. 199, pp. 63–70, Jan. 2022.

[8] A. Mahesh Pednekar, "Optimal initialization of K-means using Particle Swarm Optimization," *arXiv e-prints*, p. arXiv:1904.09098, Apr. 2019.

[9] S. Fong, S. Deb, X.-S. Yang, and Y. Zhuang, "Towards Enhancement of Performance of K-Means Clustering Using Nature-Inspired Optimization Algorithms," *The Scientific World Journal*, vol. 2014, p. e564829, Aug. 2014. Publisher: Hindawi.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[11] R. A. Fisher, "Iris." UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C56C76.

[12] B. Johnson, "Forest type mapping." UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C5QP56.

[13] V. Lohweg, "Banknote Authentication." UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C55P57.

[14] Q. Guo, Z. Yin, and P. Wang, "An Improved Three-Way K-Means Algorithm by Optimizing Cluster Centers," *Symmetry*, vol. 14, p. 1821, Sept. 2022. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

[15] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation," *Electronics*, vol. 9, p. 1295, Aug. 2020. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

[16] S. Burks, G. Harrell, and J. Wang, "On initial effects of the k-Means clustering," in *Proceedings of the International Conference on Scientific Computing (CSC)*, p. 200, The Steering Committee of The World Congress in Computer Science, Computer . . . , 2015.

[17] H. Li, B. Hu, Y. Liu, B. Yang, X. Liu, G. Li, Z. Wang, and B. Zhou, "Classification of Electricity Consumption Behavior Based on Improved K-Means and LSTM," *Applied Sciences*, vol. 11, p. 7625, Jan. 2021. Number: 16 Publisher: Multidisciplinary Digital Publishing Institute.

[18] P. E. Jebarani, N. Umadevi, H. Dang, and M. Pomplun, "A Novel Hybrid K-Means and GMM Machine Learning Model for Breast Cancer Detection," *IEEE Access*, vol. 9, pp. 146153–146162, 2021. Conference Name: IEEE Access.

[19] J. Karimov and M. Ozbayoglu, "High quality clustering of big data and solving empty-clustering problem with an evolutionary hybrid algorithm," in *2015 IEEE International Conference on Big Data (Big Data)*, pp. 1473–1478, Oct. 2015.

[20] A. Demiriz, K. Bennett, and P. Bradley, "Using assignment constraints to avoid empty clusters in k-means clustering," *Constrained clustering: advances in algorithms, theory, and applications*, vol. 201, 08 2008.

[21] N. I. Mohd Talib, N. A. Abd Majid, and S. Sahran, "Identification of Student Behavioral Patterns in Higher Education Using K-Means Clustering and Support Vector Machine," *Applied Sciences*, vol. 13, p. 3267, Jan. 2023. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[22] J. Gan, A. Li, Q.-L. Lei, H. Ren, and Y. Yang, "K-means based on active learning for support vector machine," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pp. 727–731, May 2017.

[23] G. Biau and E. Scornet, "A random forest guided tour," *TEST*, vol. 25, pp. 197–227, June 2016.

[24] Y. Ling and X. Zhang, "An Improved K-means Algorithm Based on Multiple Clustering and Density," in *2021 13th International Conference on Machine Learning and Computing*, ICMLC 2021, (New York, NY, USA), pp. 86–92, Association for Computing Machinery, June 2021.

[25] T. M. Kodinariya and P. R. Makwana, "Review on determining number of Cluster in K-Means Clustering," *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[26] S. O. Mohammadi, A. Kalhor, and H. Bodaghi, "K-Splits: Improved K-Means Clustering Algorithm to Automatically Detect the Number of Clusters," in *Computer Networks, Big Data and IoT* (A. P. Pandian, X. Fernando, and W. Haoxiang, eds.), Lecture Notes on Data Engineering and Communications Technologies, (Singapore), pp. 197–213, Springer Nature, 2022.

[27] K. P. Sinaga and M.-S. Yang, "Unsupervised K-Means Clustering Algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020. Conference Name: IEEE Access.

[28] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979. Publisher: JSTOR.

[29] A. Zhu, Z. Hua, Y. Shi, Y. Tang, and L. Miao, "An Improved K-Means Algorithm Based on Evidence Distance," *Entropy*, vol. 23, p. 1550, Nov. 2021. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[30] A. Rizwan, N. Iqbal, A. N. Khan, R. Ahmad, and D. H. Kim, "Toward Effective Pattern Recognition Based on Enhanced Weighted K-Mean Clustering Algorithm for Groundwater Resource Planning in Point Cloud," *IEEE Access*, vol. 9, pp. 130154–130169, 2021. Conference Name: IEEE Access.

[31] D. Peng, Z. Chen, J. Fu, S. Xia, and Q. Wen, "Fast k-means Clustering Based on the Neighbor Information," in *2021 International Symposium on Electrical, Electronics and Information Engineering*, ISEEIE 2021, (New York, NY, USA), pp. 551–555, Association for Computing Machinery, July 2021.

[32] M. R. Ghazi and D. Gangodkar, "Hadoop, MapReduce and HDFS: A Developers Perspective," *Procedia Computer Science*, vol. 48, pp. 45–50, Jan. 2015.

[33] H. Zhao, "Research on Improvement and Parallelization of K-Means Clustering Algorithm," in *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, pp. 57–61, Nov. 2021.

[34] Brett Lantz, *Machine Learning with R : Expert Techniques for Predictive Modeling*, vol. Third edition. Birmingham, UK: Packt Publishing, 2019.

[35] C. Timmons, A. Boskovic, S. Lakamsani, W. Gerych, L. Buquicchio, and E. Rundensteiner, "Positive Unlabeled Gradient Boosting," in *2020 IEEE MIT Undergraduate Research Technology Conference (URTC)*, pp. 1–4, Oct. 2020.

[36] G. SijiGeorgeC and B.Sumathi, "Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction," *International Journal of Advanced Computer Science and Applications*, vol. 11, 2020.