Srinakharinwirot Science Journal Vol. 38 No. 2 (2022) | 15-25

Research Article

Clustering Performance Comparison in K-Mean Clustering Variations: A Fraud Detection Study

Nattaporn Chuenjarern[1] and Subhorn Khonthapagdee[2*]

Received: 1 August 2022
Revised: 27 October 2022
Accepted: 6 December 2022

ABSTRACT

K-means clustering is a common clustering approach that is based on data partitioning. However, the k-means clustering has significant drawbacks, such as it is sensitive to deciding the initial condition. Several ways to improve the algorithm have been offered. To assess the algorithm's efficiency and correctness, the performance comparison should be evaluated. In this paper, several k-means algorithms, including random k-means, global k-means, and fast global k-means, were evaluated for their efficiency when applied to a fraud detection data set. The accuracy of each method and the Davies-Bouldin index was investigated for each algorithm to compare the clustering performance. The findings demonstrated that when a small number of groups was used, random k-means, global k-means, and fast global k-means gave similar clustering, but fast global k-means offered better errors when a big number of groups was used. Furthermore, global k-means took longer to execute than others.

Keywords: Clustering, K-Means, Global K-Means, Fast Global K-Means, BankSim Dataset

[1] Department of Mathematics, School of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

[2] Department of Computer Science, Faculty of Science, Srinakharinwirot University, Bangkok 10110, Thailand

[*] Corresponding author, email: subhorn@g.swu.ac.th

Introduction

Clustering is an idea that has been around for a long time. It is a technique for classifying data that has the same or similar characteristics to be in the same group called a cluster. The clustering is extremely valuable in many fields of science and engineering, for example, image processing, machine learning, pattern recognition, statistics, and chemical structures. More details can be found in the literatures [1-4].

The k-means algorithm is a prominent clustering approach that identifies groups by reducing the clustering error. However, the k-means algorithm is sensitive to selecting the default initial condition. It needs to select an initial guess randomly. Some researchers have proposed a new approach to deal with this trouble. The examples of proposed methods that we are interested in are global k-means and fast global k-means [5]. The global k-means is the way that dynamically inserts one cluster center with a global search technique. It consists of N k-means algorithm executions. The fast global k-means is proposed to reduce the computational complexity of global k-means.

It may also be necessary to know which algorithm performs better or is more appropriate for a given application. To analyze the algorithm's effectiveness and correctness, the performance comparison should be assessed. For example, the k-means and expectation maximization methods were examined in the literature [1] for red wine quality evaluation, and a strategy for verifying the correctness of the classification findings. In the previous study [6], the authors compared a parallel and a simple k-mean algorithm by considering the number of executions, elapsed time, and cluster quality.

Nowadays, fraud is one of the most important problems. There is an enormous case of fraud around the world. The FRAUD Magazine reported the five most scandalous frauds of 2020 that are related to various branches including financial companies, health organizations, transportation companies, and food companies. The financial company is one of the places that have a risk to occur of fraud because there are several bank transactions every single minute. So, certain transactions may involve fraud. As a result, if we can detect fraud in the transaction, the clients will be secure from financial loss. Recently, the Light Gradient Boosting Machine (LGBM) approach was proposed for accurately detecting fraudulent transactions [7]. In the literature [8], the fraud detection problem was discussed. They used data mining techniques to gain insights into the best strategy or approaches to solve a certain problem. We noticed that clustering algorithms such as K nearest neighbor and K-mean were used and showed the poor results.

Due to the lack of exploration of other variations of the k-means algorithm in the literature [8], in this work, we are interested in measuring the efficiency of several k-means algorithms, including random k-means, global k-means, and fast global k-means, when applied to the fraud detection data set. We compared the accuracy of each method capable of forming a compact cluster. We desired each cluster to be as compact as feasible while simultaneously being distinct from the others. As a result, each k-mean method was evaluated using the sum of square errors that evaluates the distance between the centroid of a cluster and each data point within that cluster. Moreover, we considered the Davies-Bouldin index or DB score that is a common approach to measure how well a k-means algorithm splits data into a specific number of clusters.

The paper was organized as follows: in the materials and methods section, we first described the process of handling the data set, k-mean algorithms, and how to build up the model. In the results and discussion section, we compared the efficacy of k-means algorithms in terms of error, DB score, and cluster density. The summary of this work was discussed in the conclusion sections.

Materials and Methods

Data gathering

We used a public bank transaction dataset named "BankSim" as in the reported literature [8]. BankSim is a simulation of bank payments based on a sample of transactional data from a Spanish bank from November 2012 to April 2013 [4]. It has 594,643 transactions, with just 7,200 of them being fraudulent. Each transaction in the dataset has several detailed features. The meaning of each feature is given in Table 1.

Table 1 The description of each attribute in the data set

Features	Description
customer	Simulated customer ID
age	Age of customer is grouped as 0,1,2,3,4,5,6, and unknown
gender	The genders are listed as enterprise, female, male, and unknown
zipcodeOri	Zip code location of customer
merchant	Simulated merchant ID
zipMerchant	Zip code location of merchant
category	15 distinct categories of transaction
amount	All prices given are in euro.
Is fraud	1 is fraudulent and 0 is non fraudulent

Data Preprocessing

Firstly, we get rid of irrelevant features or features that contain only one value, such as customer ID, merchant ID, zipcodeOri, and zipMerchant. Also, unknown age, unknown gender, and enterprise gender are dropped due to the little information and no fraud appearing in these categories. Fraud is also removed from the dataset before it is fed into the clustering analysis process. Note that this feature is used again in the evaluation and clustering profile process (Figure 1). Finally, categorical features such as age, gender, and category are converted into a format that can be used by k-means algorithms.

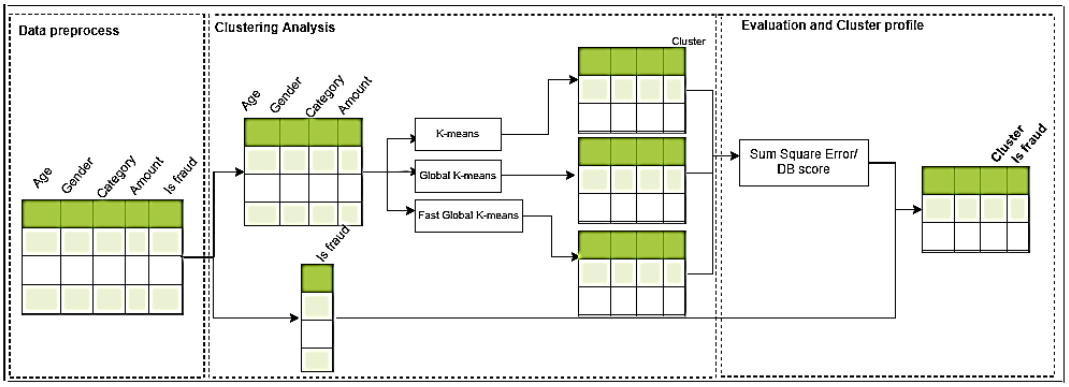

Figure 1 Overall Process

Clustering analysis

The process of clustering involves grouping or partitioning unlabeled data or objects. Analyzing the data through clustering reveals hidden patterns and structures. In clustering algorithms, Data points are clustered together in clustering algorithms so that data points in the same cluster are very similar, and data points from separate clusters are significantly distinct. This work employs the following techniques.

k-Means Clustering Algorithm

Given a data set $X=\{x_1, x_2, x_3, ..., x_N\}$ where $x_n \in R^d$ for any n = 1, 2 ,..., N. The main purpose is to separate this data set into M distinct groupings of data $C_1, C_2, C_3, ..., C_M$ called clusters in order to optimize a clustering criteria. The sum of the squared Euclidean distances between each data point x_n and the cluster center m_k to which x_n belongs is commonly used as the clustering criteria. This criterion is known as clustering error, and it is determined by the cluster centers $m_1, m_2, m_3, ..., m_M$. The criteria is defined by

$$E_{sum} = E(m_1, m_2, m_3, ..., m_M) = \sum_{i=1}^{N} \sum_{k=1}^{M} I(x_i \in C_k) \, ||x_i - m_k||^2 \quad (1)$$

where $I(X)$ = 1 if X is true and 0 otherwise. With respect to the clustering sum error, the k-means process achieves locally optimum solutions. The method's major downside is its sensitivity to the cluster center's initial position. The algorithmic steps for the K-mean algorithm are listed below.

Algorithm : k-Means Clustering Algorithm	
Input	No. of clusters k and Set of N data points $X = \{x_1, x_2, x_3, ..., x_N\}$
Output	A set of k clusters
Step 1	Select k data points at random from X to initialize the k centroids.
Step 2	Calculate the distance between each data object x_i $(1 \leq i \leq N)$ and each of the k clusters C_j $(1 \leq j \leq k)$, and then allocate the data object to the cluster that is closest to it.
Step 3	Recomputed the centroid of each cluster by evaluating the mean of all the data points in each cluster
Step 4	Repeat **Step 2** until centroids do not change

Global k-Means Clustering Algorithm [5, 9]

To overcome the K- mean algorithm's fundamental issue, which is its sensitivity to the initial locations of the cluster centroids, the global k-means clustering algorithm has been proposed. It is a deterministic global optimization approach that uses the k-means algorithm as a local search process and does not rely on any initial parameter values. The method takes an incremental approach, adding one new cluster center at a time rather than choosing starting values at random for all cluster centers. The algorithmic steps for the global k-mean algorithm are listed below.

Algorithm : Global k-Means Clustering Algorithm	
Input	No. of clusters k and Set of N data points $X = \{x_1, x_2, x_3, ..., x_N\}$
Output	A set of k clusters
Step 1	For k = 1, Compute the centroid m_1 by calculating the mean of all the data point in X.
Step 2	Set k = k+1. Consider the centroid $\{m_1, m_2, m_3,...,m_{k-1}\}$ and use the data point x_i $(1 \leq i \leq N)$ as the initial k^{th} cluster center.
Step 3	Execute the k-means algorithm N times. The best solution achieved after the N executions is regarded to the solution for clustering problem with k = k+1.
Step 4	Repeat **Step 2**

Fast Global k-means Clustering Algorithm [5]

The global k-means clustering technique in X takes N executions. As a result, the computational cost of the Global k-means method is somewhat larger. To speed up the execution, the fast global k-Means clustering algorithm was proposed. It does not repeat the k-Means procedure for each data point in order to solve the k−clustering problem. Instead, the algorithm computes the upper bound $E_{sum,i} \leq E - b_i$ on the resulting error $E_{sum,i}$ for each possible data point x_i, where E is the error value of (k-1)-clustering problem and b_i is defined as:

$$b_i = \sum_{j=1}^{N} \max\left(d_{k-1}^{j} - ||x_i - x_j||^2, 0\right), \qquad \forall x = 1, 2, ..., N. \tag{2}$$

Here, d_{k-1}^{j} is the squared distance between x_j and the nearest cluster center found so far among the (k-1) cluster centers, that is, the squared distance between x_j and the center of the cluster to which it belongs. Here, we want x_i, that minimizes $E_{sum,i}$ which is equivalent to x_i, with the largest b_i. This data point x_i, will be collected to be the initial k^{th} cluster center.

Algorithm : Fast Global k-Means Clustering Algorithm	
Input	No. of clusters k and Set of N data points $X = \{x_1, x_2, x_3,..., x_N\}$
Output	A set of k clusters
Step 1	For k = 1 Compute the centroid m_1 by calculating the mean of all the data point in X.
Step 2	For k = k+1. Compute b_i for $1 \leq i \leq N$.
Step 3	Use the data point x_i with the largest b_i as the initial k^{th} cluster center.
Step 4	Execute the k-Means algorithm to obtain the solution for clustering problem with k = k+1.

K-means clustering, global k-means clustering and fast global k-Means clustering algorithm are clustering methods based on data partitioning. The global k-means clustering algorithm is independent to initial conditions. It provides great results in terms of sum of squared errors. The method performs the k-Means algorithm with several random restarts. However, it may take a huge computational cost. The fast global k-means provides similar results to the global k-means method, but it is much more robust than the global k-means algorithm.

Model set up

Each algorithm was executed with a k ranging from 2 to 140 (due to the computation cost). The algorithm would continue to run until the difference between the prior and current errors were less than 0.001. The centroid initialization in random k-mean was selected at random from the dataset. Because of the randomization, we run the random k-mean algorithm experiment 5 times and report the average error. We recorded errors, centroid lists, and cluster labels after each execution. We also reorganize cluster labels based on their size after gathering all findings. The smaller the size, the higher the label. Principal component analysis (PCA) was used to reduce the higher dimension to two dimensions in order to see the location of the centroid in each model. In this study, we use PCA that is implemented in scikit-learn library.

Evaluation

In this study, two metrics were utilized to assess the performance of each algorithm. 1) Sum of Euclidean distances squared or sum of squared error (SSE). Clustering error in equation (1) is used in this work. It computes the difference between a cluster's centroid and each data point contained inside that cluster. In other words, it determines how dense a cluster is packed. Similar data should be as close to each other as possible. In general, the objective of the k-mean algorithm is to reduce clustering errors, therefore, the lower the better. 2) The Davies-Bouldin index, often known as the DB score, is a proportion of cluster error to cluster separation error. A lower score indicates more successful clustering. Let s_i and s_j be the average Euclidean distance between each point of the cluster and the centroid in cluster i and cluster j respectively. Let d_{ij} be the distance between cluster centroids i and j. Then we have a Davies-Bouldin Index for a given pair of clusters i and cluster j, R_{ij}, is defined as

$$R_{ij} = (s_i + s_j) / d_{ij} \quad (3)$$

Hence the Davies–Bouldin index is defined as:

$$DB = 1/k \sum_{i=1}^{k} \max_{i \neq 1} R_{ij} \quad (4)$$

Results and Discussion

We observed that the overall trend of SSE, DB score, and centroid position from random k-mean, fast global k-mean, and global k-mean are quite comparable. In particular, Figure 2 illustrates that when the number of k is small, for example, k = 2 to k = 8, the errors of random k-mean and fast global k-mean are similar but somewhat higher than global k-mean. If k > 10, the error of the fast global k-mean and the global k-mean trend are less than that of the random k-mean. When k > 40, the fast global k-mean algorithm has a rather lower error, as demonstrated in **Figure 3**. However, we were able to execute the model from k = 2 to k = 40 since the global k-mean algorithm involves time-consuming computation.

The overall trend of the DB score from each k-mean is very comparable, with the exception of k = 4, where the DB score of random k-means is surprisingly rather high. However, as k increases, the score increases, and when k exceeds 20, it approaches 1. Moreover, only k = 2, 3, 4, 5, and 8 have DB scores less than 0.6, with k = 2 having the lowest DB score (Figure 4).

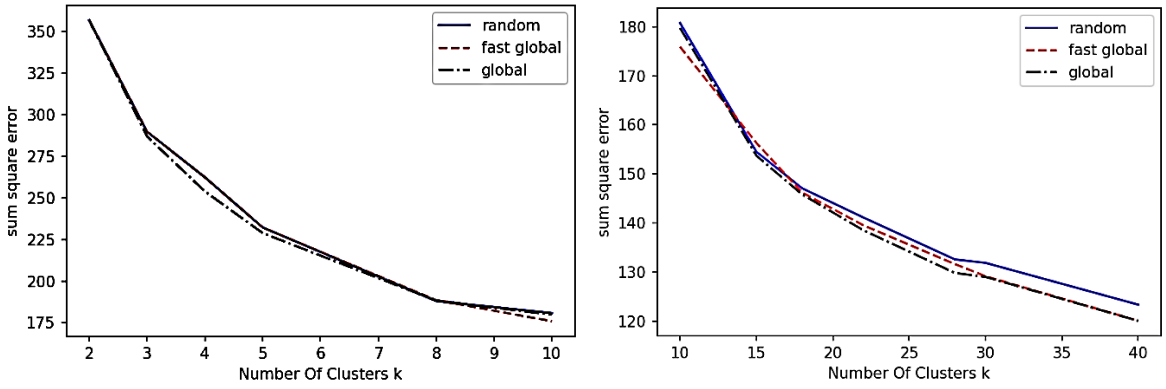

Figure 2 The sum square error of each method for k = 2 to k = 10 (Left). The sum square error of each method for k = 10 to k = 40 (Right).

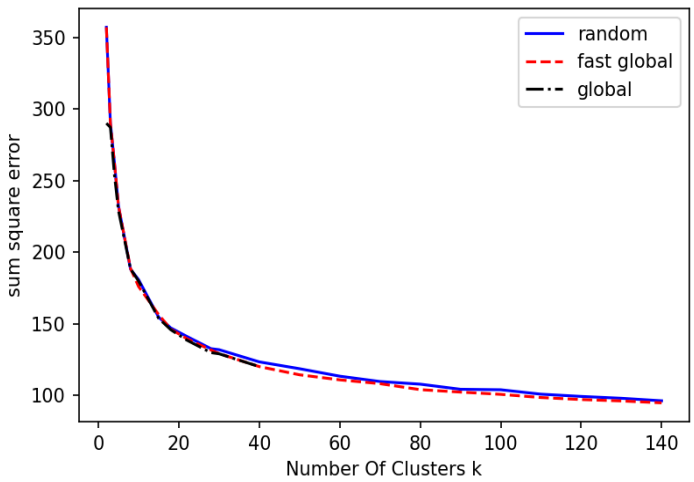

Figure 3 The sum square error of each method for k = 2 to k = 140.

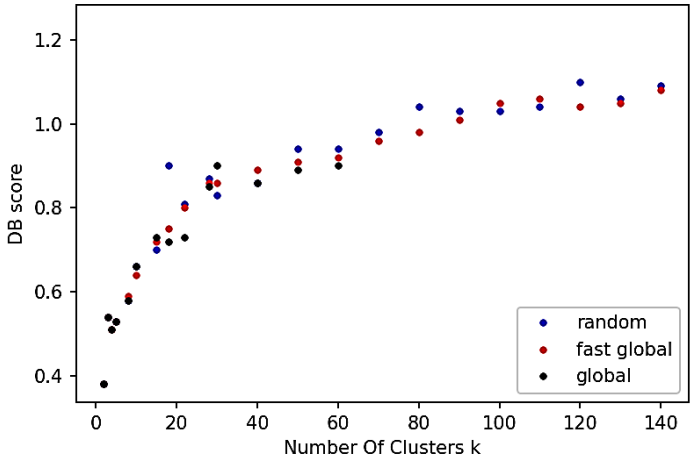

Figure 4 The DB Score of random k-mean, fast global k-mean, and global k-mean.

By using PCA algorithm that we mentioned in the Model setup section, we were able to transform centroids into two dimensions. The average of the centroids from each algorithm was then used to generate a reference centroid. The average of the centroids from each algorithm was then used to create a reference centroid. Then, utilizing these centroids, we computed a mean of difference. In the case of $k < 8$, the mean difference ranges from 0.4 to 1. When we considered the larger k, the mean difference increased slightly (Figure 5).

Finally, we examined each cluster to gain a better understanding of how the data in each cluster behaves. Each cluster from various algorithms had a comparable overall size and trend. There is always a majority and a minority cluster. For example, cluster 0 in $k = 2$, cluster 0, 1 in $k = 3$, 4 and cluster 0, 1 and 2 in $k = 8$ are majority clusters. As K increases, the majority cluster splits into a small number of clusters of the same size, for example, clusters 0 and 1 in $k = 5$ and clusters 0, 1 and 2 in $k = 8$. We also notice that the minority clusters are associated with more fraudulent data (see Figure 7). Since each transaction in the dataset was labeled as fraud or non-fraud, we were able to compare the amount of fraudulent data in each cluster. The smallest cluster frequently has the highest percentage of fraudulent data and, unsurprisingly, the highest range of transaction amounts. From Figure 7 and Figure 8, it is clear to see that the majority cluster contains most of the non-fraud data that has a lower range of transaction amounts.

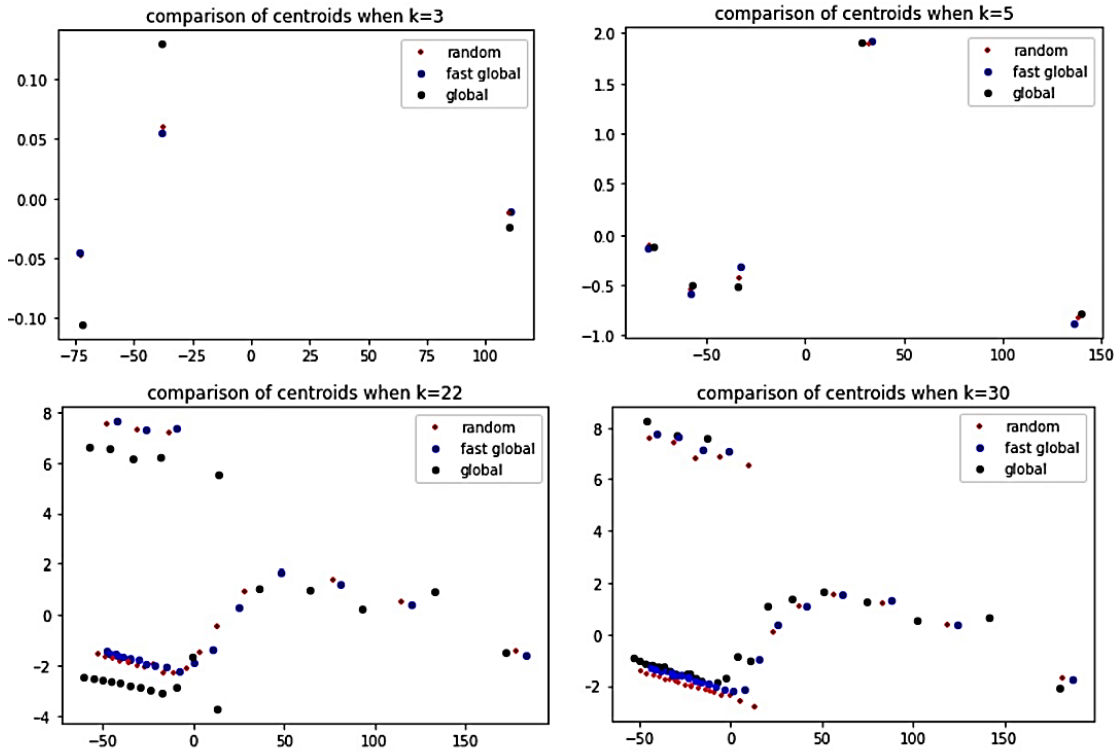

Figure 5 Comparison of centroids when k = 3, 5, 22 and 30.

Apart from the majority cluster (fraud cluster) and minority cluster (non-fraud cluster) under the scenario where k > 2, we can refer to those clusters in between as suspicious clusters. For example, k = 3 in Figure 8 shows that the range of transaction amount is higher than data in the non-fraud cluster. However, the percentage of fraud in this cluster is only 22 %.

Unfortunately, there is no algorithm in our study that can completely distinguish between fraud and normal data. The best cluster that has the highest percentage of fraudulent data is cluster 1 in k = 2 from the fast global k mean. It contains 80% of total fraud. The accuracy of overall fraud and non-fraud is 0.97. When k is equal, the smallest cluster from the fast global k mean always contains the highest percentage of fraudulent data compared to other algorithms.

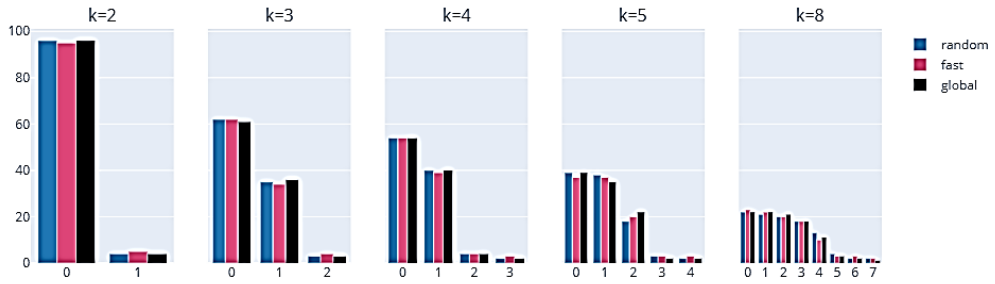

Figure 6 Comparison of size percentage of each cluster when k = 2-8.

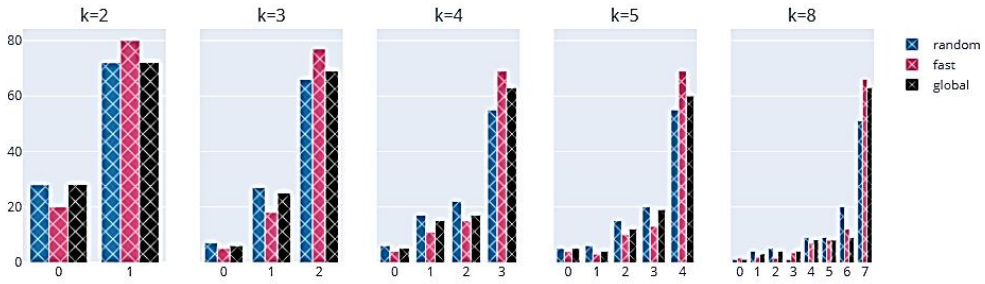

Figure 7 Comparison of percentage of fraudulent data from each cluster when k = 2-8.

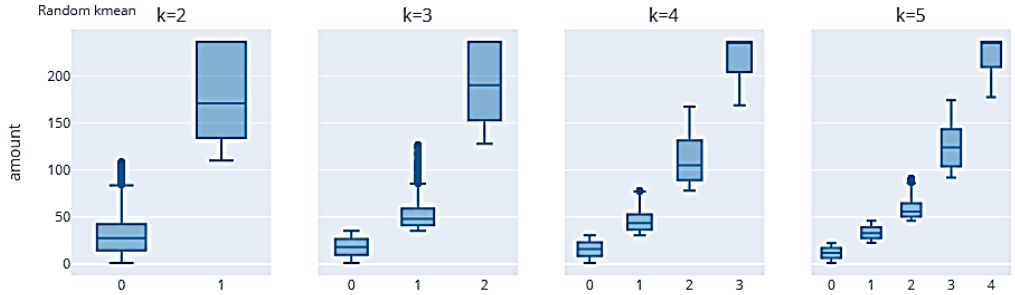

Figure 8 Distribution of transaction amount of each cluster when k = 2-5 from random K mean algorithm.

Conclusions

In this paper, the efficiency of random k-means, global k-means, and fast global k-means was measured using the Banksim fraud detection data set. To compare the accuracy of each method capable of forming a compact cluster, the sum of square errors and the Davies-Bouldin method were used. When the number of clusters were not large, the results showed that their errors were not significantly different. When the number of clusters was greater than 40, the fast global k-mean algorithm had a slightly lower error. However, the results revealed that clustering with a small number of clusters was more successful based on the DB score. Despite the fact that each algorithm used a different method to find the centroid, when the number of clusters was increased for the centroid comparison, the centroids only differed slightly. This led to an overall trend of cluster size and percent fraud in each cluster that was almost the same. We also provided some detail about clustering profile especially the distribution of fraud data and transaction amount. Following the completion of the studies, these three k-means algorithms provided results that were slightly comparable, but they differed in terms of computational cost. We may apply random k-means on the data set to save time. If a more precise conclusion is required, we can use the fast global k-means method. For future work, it is interesting to further examine the distribution of the age of customers and the category of transaction in each cluster to give a full cluster profile. Moreover, other types of clustering algorithms such as DBSCAN will be studied.

Acknowledgements

The authors would like to thank the referee for reading our work thoroughly and providing helpful comments and ideas that improved the quality of this paper in its current version. The authors would also like to express their gratitude to Woraphat Diaosurin, Tawatchai Amatayakul, and Wachirawit Pipitpuwasrikul for igniting the concept for this study in their senior project.

References

1. Jung YG, Kang MS, Heo J. Clustering performance comparison using k-means and expectation maximization algorithms. Biotechnol Biotechnol Equip. 2014 Nov 14;28(Sup1):S44-8.
2. Stahl M, Mauser H, Tsui M, & Taylor NR. A robust clustering method for chemical structures. J Med Chem. 2005 Jun 2;48(13):4358–66.
3. Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognit Lett. 2010; 31(8):651-66.
4. Xu R, Wunsch D. Survey of clustering algorithms. IEEE Trans Neural Netw. 2005 May;16(3):645-78.
5. Agrawal A, Gupta H. Global k-means (GKM) Clustering algorithm: a survey. Int J Comput Appl. 2013 Oct;19(2):20-4.
6. Islam Z, Shaukat A, Inayat K, Myriam H, Hela E, Muhammad Z, et.al. Performance evaluation of simple k-mean and parallel k-mean clustering algorithms: big data business process management concept. Mob Inf Syst. 2022 Jun 23;1-15.
7. Aziz RM, Baluch MF, Patel S, Ganie AH. LGBM: a machine learning approach for Ethereum fraud detection. Int J Inf Technol. 2022 Dec;14:3321-31.
8. Dabab M, Freiling M, Rahman N, Sagalowicz D. A decision model for data mining techniques. Proceedings of Portland International Conference on Management of Engineering and Technology. 2018 Aug 19-23;Honolulu., Hawaii; 2018. p. 1-8
9. Likas A, Vlassis N, Verbeek JJ. The global k-means clustering algorithm. Pattern Recognit. 2003 Feb;36(2):451-61.