Fast and Memory-Efficient Implementation of Gaussian Mixture Model for Big Data with Application to Satellite Images

Ishfaq Ali[1,3*], Atiq Ur Rahman[2], Kerrie Mengersen[1*], Dost Muhammad Khan[3]

[1*]School of Mathematical Sciences and Center for Data Science, Queensland University of Technology, 2 George St, Brisbane, 4000, QLD, Australia.
[2]Collage of Business and Economics, Australian National University, 26C Kingsley Street, Canberra, 2601, ACT, Australia.
[3]Department of Statistics, Abdul Wali Khan University, Mardan, 23200, KPK, Pakistan.

*Corresponding author(s). E-mail(s): ishfaqali8380@gmail.com; k.mengersen@qut.edu.au;
Contributing authors: atiqur.rehman@anu.edu.au; dostmuhammad@awkum.edu.pk;

Abstract

The expanding complexity and volume of remote sensing data require new techniques for efficient processing and accurate object detection within images. In geospatial communities, satellite images serve as a crucial tool for the exploration and analysis of diverse features like dynamic changes in forests or urbanization patterns over time, detection and control of forest fires, categorization of land use, assessment of crop damage and growth, etc. However, for big datasets, the Gaussian Mixture Model is computationally expensive. In this paper, we propose a new algorithm, which is memory-efficient and computationally less expensive. The proposed algorithm utilizes $\boldsymbol{K}$-means to extract useful information from complex data, known as quantized data and then uses the Gaussian Mixture Model to refine the clustering results. The proposed algorithm is testified on both the real-world and synthetic datasets. The maximum scores of Dunn and Silhouette indexes of the proposed method in the real-world datasets keep a more optimal balance between the model fit and complexity. The proposed algorithm is then

evaluated for the four different synthetic datasets with their quantized datasets where the parameters such as clusters, means, variances, BIC and shapes of both the datasets are compared with the true parameters. So, it is concluded that quantized data can be used as a proxy for the larger datasets. Similarly, the real world application of the proposed method has also been applied on the satellite images of Islamabad where the algorithm identifies different clusters correctly such as water, forest, urbanization and types of land use.

Keywords: Big data; Remote sensing; Unsupervised Clustering; ***K***-means; Gaussian Mixture Model; Quantized data; Satellite images.

1 Introduction

The growth of high-resolution satellites during the recent years has resulted in a significant increase in the volume of satellite images which in turn has accelerated the field of image processing. This acceleration has been further enabled by advances in big data analysis and cloud computing. In geospatial communities such as agricultural, environmental monitoring and management, demography and public health, satellite images are used as a tool for investigating and analyzing features such as changes in forest or urbanization over time, detection and control of forest fires, types of land use, crop damage and growth, etc. However, identifying such phenomena visually in remote-sensing images can be challenging. Unsupervised clustering techniques can be used as an automated alternative for analyzing the satellite images and different phenomena of interest [1].

Unsupervised clustering methods hold a well-established significance within the realm of statistics and machine learning literature. Many unsupervised classification techniques have been developed to effectively handle extensive volumes of data. Among these techniques, the K-means clustering algorithm stands out due to its simplicity and efficiency, especially when dealing with large datasets. As a result, it remains a popular and widely utilized approach across various domains, including remote sensing image segmentation [2]. However, the K-means algorithm does come with certain limitations. A prominent drawback is its classification nature known as hard clustering, where each data point is unequivocally assigned to a single cluster. Additionally, the algorithm operates under the assumption that clusters exhibit a spherical shape and possess equal variances across all clusters—an assumption that often diverges from reality in many real-world datasets. Furthermore, it can be expensive in terms of memory usage, potentially leading to poor performance in terms of segmentation accuracy, particularly concerning satellite images. A popular alternative is the Gaussian Mixture Model (GMM) [3, 4]. This approach retains the appeal of K-means in terms of conceptual, simplicity and flexibility but leads in performance over K-means due to its probabilistic nature (soft clustering). Therefore, it is widely adopted as a data-driven way to identify clusters in the data. The traditional

expectation-maximization (EM) algorithm [5] can be used to estimate the parameters of the Gaussian Mixture Model. However, this algorithm may be computationally expensive when dealing with big datasets such as Landsat imagery [6].

To address this issue, researchers have proposed various techniques to improve the computational efficiency of the EM algorithm for big data [7], such as data-parallel programming model [8, 9] and graphics processing units (GPU) [10]. The fast robust expectation maximization (FREM) algorithm addresses these issues by introducing robust techniques and modifications to the traditional EM algorithm such as using a robust estimator for the mean and covariance matrix and incorporating a regularization term to prevent zero variances [11]. To improve the computational efficiency of the Gaussian Mixture Model, various strategies have been proposed by Verma et al., [12], such as grid-based sampling methods, adaptive methods and parallel computation methods. However, these methods are limited to data in two and three dimensions and need to be further optimized to handle big data. Neagoe et al., [13] proposed a technique by combining the two clustering algorithms, the K-means and the GMM. The authors, at first, implemented the K-means to initialize the key parameters such as mixing coefficient, means, and covariance matrices for the GMM-EM algorithm. In the second step, they put the initialized parameters into the GMM-EM algorithm as initial value to exceed the convergence of EM algorithm. However, the weak initialization of the parameters can lead to suboptimal solutions or it can converge to local optimal instead of the global optimum. The wrong initialization of parameter can also lead to slow convergence which requires more iterations for the algorithm for stabilization. Besides, if the initial parameters lead to rather small or rather large covariance matrices, then, as a result, the GMM-EM may either over fit or under fit the data.

Numerous researchers have adopted approximate Bayesian inference techniques e.g variational inference [14, 15] and approximate Bayesian computation [16, 17]. These methods have been successful in reducing the computational burden of the EM-GMM algorithm and improving its performance in big data applications. Guha et al., [18] proposed a clustering algorithm to handle large datasets, known as CURE (Clustering Using Representatives). CURE is based on a hierarchical technique and utilizes multiple representative points in each cluster. Another approach to handling big data in clustering is using sampling-based algorithms such as Clustering Algorithm based on Randomized Search (CLARANS) [19] and Clustering for Large Applications (CLARA) [20] which divide the data into smaller subsets and perform the clustering in parallel on each subset then combine the results to produce the final clustering. Such algorithms are not effective when the data sizes exceed the memory sizes. The Gibbs sampling method can be a promising approach to estimate the parameters of Gaussian Mixture Model [21]. However, it can be computationally intensive and may require a large number of iterations to converge. Huang and Gelman [22] proposed the divide-and-conquer strategy that randomly divides the data into non-composite subsets and retrieves samples from the posterior given subset by adopting the Markov chain Monte Carlo (MCMC) algorithm independently on each separate subset. The use of the MCMC algorithm helps to overcome the limitation of the EM algorithm, making it more suitable for big data applications when applied efficiently. Additionally, the use of Sequential Monte Carlo sampling to guide the selection of the components of

interest by analyzing the initial sub-sample has also been proposed as a way to enhance computational power [23]. An adequate representation of the interesting component in the initial random sample is necessary to make it applicable. Other alternative method for data reduction include symbolic data that involves the transformation of a vast dataset of individuals into classes, thereby diminishing its overall size and retaining as much information as possible (for example see [24, 25]). It is important to note that each method has its own advantages and disadvantages and the choice of which method to use will depend on the specific problem and dataset being considered.

Apart from the above discussion, there are several deep learning approaches available in the literature to overcome the complexity of time and have haul cynosure attention at analyzing the big data. The SpectralNet as proposed by Shaham et al., [26] is an epitome that the authors did by making deep learning combination with the spectral clustering and it overcomed the time complexity issue exquisitely in spectral clustering. The Fully Convolutional Network (FCN) [27] is also one of the most important works in deep learning for semantic segmentation. Similarly, the Convolutional Neural Networks (CNNs) are widely utilized in the field of remote sensing for investigating features such as forest, urbanization, and types of land use [28]. For brief summary of CNNs, we recommend the book written by Goodfellow and colleagues [29]. But inspite its potential, Deep Learning (DL) has been facing severe hurdles, particularly in developing countries, i.e, Pakistan due to the limited resources platforms [30]. For example, in the fields of medical images or remote sensing, may be questioned deep learning for their costs (global tasks) because of the shortage of sufficient training data [28, 31, 32]. The author Han et al., [33] asserted that the high cost of data acquisition and lack of computing hardware such as CPU, GPU, and FPGA did not allow DL models magnificently. Besides, the DL models, particularly the Deep Neural Networks (DNNs), are often considered as "Black Boxes" because they can not interpret their decisions vividly and the user can not understand the referenced results easily [34].

In large datasets, it is observed that the input information is similar in nature for example in satellite imagery parks and playgrounds are largely identical in a specific area except for some noise data like a unique feature that makes a park different from others [35]. Likewise, images of huge reservoirs of water also contribute to millions of repetitive observations. Large bodies with similar visual attributes are over-sampled by the sampling-based approaches, which tends to the smaller clusters of interest (vital information) being ignored. Consequently, this may end up in poor-quality clusters since the sampling results tend to be biased towards a minute or less number of huge clusters [36]. A wise approach known as data quantization by averaging similar components can be useful in reducing the size of the dataset and removing redundant information which in turn reduces the computational burden of the clustering algorithm. Additionally, data quantization helps to identify the most important and unique characteristics of each cluster making the clustering results more robust and accurate. However, it is important to note that data quantization may also lead to loss of information and should be used with caution.

Motivated by image segmentation and recent research works on the scaling of clustering approaches to big datasets, we propose a new algorithm that combines

both K-means and the Gaussian Mixture Model which is capable of fitting the data distribution with great precision and partitioning the data that closely match their expected clusters. The proposed algorithm first uses the K-means clustering algorithm to shrink the data and then uses the Gaussian Mixture Model to refine the clustering results. The proposed approach can effectively reduce the impact of noise on the segmentation outcome, making it a suitable solution for big data clustering. The proposed method gains significant improvement from days to hours in processing time.

The rest of the article is arranged as follows. A detailed explanation of the proposed method is elaborated in section 2. Section 3 and 5 include results and a discussion where the performance and effectiveness of the proposed method are evaluated based on the simulations and real-world application. The conclusion is given in section 6.

2 Methodology

2.1 Gaussian Mixture Model

Let $X = \{x_1, x_2, ..., x_n\}$ be a vector of real-valued observations in a d-dimensional space. A Gaussian Mixture Model (GMM) is the weighted sum of the M Gaussian probability distributions and can be defined as follows:

$$\mathcal{P}(x \mid \psi) = \sum_{k=1}^{M} \pi_k \mathcal{N}(x \mid \psi_k = \{\mu_k, \Sigma_k\}), \tag{1}$$

where, $\psi = \{\pi_k, \psi_k\}$ for $k = 1, 2, \ldots, M$, and $\mathcal{N}(x|\psi_k)$ denote the multivariate Gaussian distribution with parameter ψ_k. The mixing proportion π_k is the prior probability of x associated with the k^{th} Gaussian component; it naturally satisfies $\sum_{k=1}^{M} \pi_k = 1$ and $0 \leq \pi_k \leq 1$.

The maximization of the likelihood function for the estimation of the parameter vector, ψ, is not feasible. However, the expectation-maximization (EM) algorithm [5] framework can be utilized to compute the maximum likelihood estimate (MLE) of the model parameters when dealing with incomplete or missing data. An overview of the EM algorithm is provided in Appendix A.

2.2 Data Reduction Strategy

In large-scale datasets, when the large segments of the data comprise similar information, a wise approach would be to cluster homogeneous observations together to acquire a suitable representation from each cluster. However, this approach may result in a significant loss of information, which can be alleviated by incorporating a moderately substantial number of clusters. The word "moderately substantial" is utilized to emphasize the fundamental trade-off between the number of clusters and the potential loss of information; a smaller number of clusters leads to a maximum loss of information. A quantization step (rather than sampling) is used in the first-level clustering, aiming to replace a larger set of values onto a smaller set by reducing the impact of noise.

We accomplish the aforementioned task using the widely adopted K-means [37] clustering algorithm, known for its scalability and efficiency in big datasets. Numerous algorithms have been put forward to address the K-means problem, but it is worth noting that the algorithm described in [38] is well-known for its excellent performance. The algorithm utilizes a similarity matrix based on Euclidean distance, to minimize the sum of squared distances between the observations in each cluster and their respective cluster centers:

$$d(x, c) = \sqrt{\sum_{x_i \in c_k} (x_i - \mu_k)^2} \tag{2}$$

where x_i denotes a data point within the cluster c_k and μ_k represents the mean value of the points assigned to the same cluster c_k.

To maintain a close resemblance between the quantized set and the original dataset, we utilize a large number of clusters. We evaluate the ratio between within-cluster sum of squares (WCSS) and the total cluster sum of squares (TCSS) to determine the large number of clusters. When this ratio approaches zero as given in equation 5, it signifies the attainment of an optimal clustering solution. That is

$$\text{WCSS} = \sum_{g=1}^{k} \sum_{i=1}^{n} (x_{ig} - \bar{x}_g)^2, \tag{3}$$

$$\text{TCSS} = \sum_{g=1}^{k} \sum_{i=1}^{n} (x_{ig} - \bar{x})^2, \tag{4}$$

$$\frac{\text{WCSS}}{\text{TCSS}} \rightarrow 0, \tag{5}$$

where

- x_{ig} represents the i^{th} data point in the g^{th} cluster.
- $\bar{x}_g$ is the mean (average) value of the data points in the g_{th} cluster.
- $\bar{x}$ is the overall mean (average) of all the data points.

Once the clusters are formed, a mean (quantized) value of each cluster can be chosen. This mean can be thought of as a "compressed" version of the cluster, containing the most important or salient features of all the data points in the cluster. It might also be worth noting that the average/ means is the first movement, so is the most informative first summary statistic for the cluster. Through this approach, we effectively sidestep two common issues (convergence to a local optimum and model selection) inherent to K-means clustering. Fig. 1 portrays a simulated dataset consisting of 10,000 observations (grey-colored) generated from a 5-component Gaussian mixture, with an overlay of 500 quantized values (red-colored) obtained through K-means clustering. It is important to keep in mind that we apply the K-means algorithm as a step of preliminary dimension reduction because it has the potential to allay the computational burden for the flexible and sophisticated mixture models. Furthermore, this approach permits the integration of identical information and also keeps an eye on the correlation between variables.

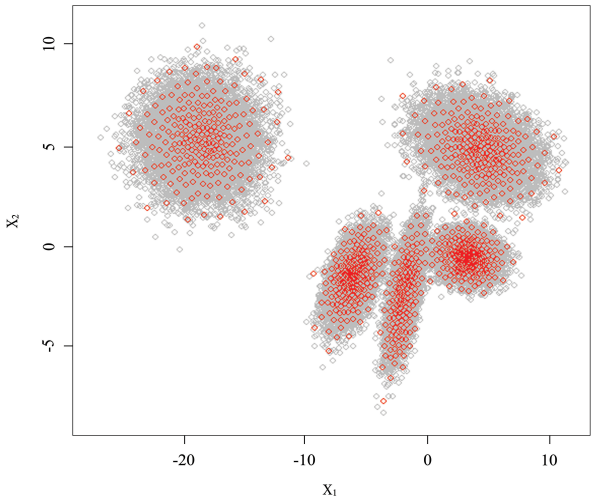

Fig. 1: The grey and the red points depict the original and the quantized dataset, respectively.

2.3 GMM for Big Data with Reduced Representation

As mentioned earlier, the conventional expectation-maximization (EM) algorithm used to estimate the Gaussian Mixture Model parameters is computationally expensive when applied to big datasets. To enhance the utility of a Gaussian Mixture Model when dealing with large datasets, one approach is to decrease the computational burden by reducing the size of the dataset. This can be accomplished by utilizing the first step, where the traditional K-means clustering algorithm acts as a summarization technique (say, quantization method) that generates an in-core representation of the data. In this way, the original dataset size N, is reduced to an informative smaller dataset of size k. Depending on its complexity and size, using a small subset of the data, such as 5 to 10% of the total amount of data, can indeed be an effective way to save time and improve the performance of a clustering algorithm.

In the second step, the reduced representation of the data is used as input for GMM. This process decreases the number of clusters from k to $k_0 \ll k$. The gradient function [39] is used to make the number of clusters. The gradient of the BIC score provides information on how changes in model parameters influence the balance between model fit and complexity. The point where the BIC scores start to level off or decrease more slowly which is often considered the optimal number of clusters. We utilize the Bayesian Information Criteria (BIC) [40] to identify an optimal number of clusters. The BIC score is computed as follows:

$$BIC = -2\ell_m(x, \hat{\theta}) + p \cdot A_n \tag{6}$$

where the term $l_m(x, \hat{\theta})$ denotes the log-likelihood of the entire dataset under the model, the symbol $\hat{\theta}$ represents the parameter values that maximize the likelihood function, p represents the number of independent parameters in the model and $A_n = log(n)$, n denotes the number of data points x. The BIC score not only indicates the number of groups but also describes the shape and volume of the respective groups. Different Information Criteria are discussed in section 3.

2.4 Algorithm

The proposed algorithm is outlined in the following steps.
Input: $\boldsymbol{X}$, an array of size $P \times N$ where P and N denote the number of variables and observations, respectively, in a large dataset.
Output 1: $\boldsymbol{Q}$, an array of size $p \times n$ where p and n denote the number of variables and observations, respectively, in the quantized dataset.
Output 2: Determine the number of clusters in the quantized dataset using GMM.
Choose small sizes of the quantized dataset, ranging from 5% to 10% of the large dataset.
step-1: For i in 1 to m, where m is the number of quantized datasets.
step-2: Choose $i \ll N$ clusters using K-means clustering algorithm.
step-3: Calculate the ratio of within and total sum of squares of i^{th} clusters.
step-4: Choose $\boldsymbol{Q}$ such that the value described by equation 5 approaches zero and remains constant as we increase the index i.
step-5: Utilize GMM on quantized dataset. This approach will decrease the number of clusters from m to k.
step-6: Select the number of components based on equation 6.
step-7: Extend the number of components to a large dataset.

3 Simulation Study

This section includes a description of a set of simulation datasets, the implementation of our proposed technique and the desired results. Many researchers have used synthetically generated datasets to validate the results of clustering algorithms [41], as these datasets have known properties that allow for easy evaluation of the algorithm's performance. Franti and Seinaroja [42] proposed the use of benchmark datasets to study the statistical analysis of clustering algorithms. They argue that using real-world datasets can be problematic as it is often unclear what the true partitioning structure of the data is and the ground truth is not known. Therefore, using benchmark datasets can provide more accurate results as they have known properties and ground truth that allow for a more rigorous evaluation of the clustering algorithm performance. Thus, synthetic datasets are a useful tool to evaluate the performance of clustering algorithms and to prepare them for real-world applications.

3.1 Datasets and Experimental Procedure

For the simulation studies, four synthetic datasets are randomly generated from M Gaussian distribution. Each dataset is characterized by the parameters; for instance,

number of clusters, number of points, number of dimensions, weight parameter, standard deviation, σ and mean, μ. These parameters are used to generate the synthetic datasets, with the assumption that each point in the dataset is independently generated from one of the M Gaussian distributions. Table 1 provides a summary of the characteristics of the synthetic datasets used in the experiments. The datasets are generated using the "VVV" model (VVV denotes that the volume, shape, and orientation of the clusters vary), where there are 50,000 data points in each dataset with different cluster sizes.

Table 1: Brief description of the datasets

Datasets	Number of data points	Number of clusters
S1	50,000	2
S2	50,000	3
S3	50,000	5
S4	50,000	7

The "K-means()" function in the "stats" package of the R programming language is used for the first stage of the approach and to obtain the quantized values. To improve the accuracy of the K-means clustering algorithm, we used five random starting points in this study. It is to keep in mind that the K-means() function provides the option to specify multiple random starting points. However, a large number of starting points will increase the computation cost due to the iteration of the algorithm. To minimize the computational cost, the "foreach" package in R is used to run the K-means algorithm in parallel with a "foreach" loop. In the second stage of the Gaussian Mixture Model is fit to the quantized data using the "mclust" package in R [43]. The "mclust" package provides functions for model-based clustering, classification and density estimation based on finite Gaussian mixture modeling. The study is carried out on a computer with specification Inter (R) Core (TM) i7-5500U CPU @ 2.40 GHz; RAM 16 GB with Windows 10, 64-bit Operating System.

3.2 Experimental results and discussion

In this study, the proposed method is applied to both the quantized and original datasets to evaluate its efficiency. The results obtained from the quantized datasets, including means, probabilities, variances, shape and volume are compared with those obtained from the original datasets. To accurately represent the structure of the data, large quantized datasets are used in the study. The use of large datasets helps to capture the resolution of big data while still being small enough to remove computational barriers. An in-depth search operation is performed through the K-means clustering algorithm to determine the best subset (quantized dataset) from the entire dataset. The quantized data that are obtained through the K-means clustering algorithm perform well on a single run. The appropriate value of a quantized dataset can be achieved by stopping criterion as described in section 2.2. It is anticipated that

significant improvement in the competitive performance of the quantized datasets will be obtained by increasing their sizes. However, it is important to note that there is a trade-off between the quality of the clustering solution and the computational cost of the algorithm. As the number of clusters increases, the algorithm will require more computational resources and time to converge.

In the study, five different quantized dataset sizes (100, 500, 1000, 1500 and 2000) are compared to the original dataset to see how the size of the data affected the results. A preliminary analysis of original and quantized data using correlation and Q-Q plots is shown in Fig. 2. The Q-Q plot is a powerful graphical tool for assessing the similarity between two probability distributions. It estimates the quantiles of two datasets which shows whether the two datasets come from populations with a common distribution or not.

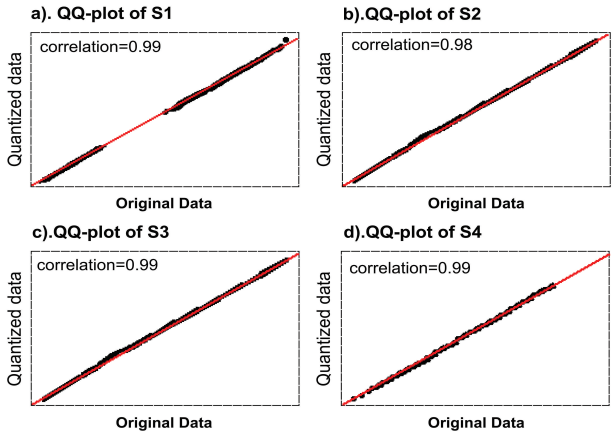

Fig. 2: Visual comparison of the quantiles of original dataset vs quantized values.

Fig. 2 shows that both datasets originate from populations with a common distribution because the points on the Q-Q plot fall roughly on a straight line. Moreover, the correlation between the two sets of points exceeds 0.98 in all cases. The quantized datasets therefore represent the original dataset well and can be used as a proxy for the larger datasets. It is observed that the Q-Q plot is not affected by different sizes of quantized data.

The performance of the GMM is evaluated by using both the quantized datasets (range from 100 to 2000) and original datasets the results are depicted in Table 2. It is found that the appropriate sizes for the quantized values of the datasets S1, S2, S3 and S4 are 1000, 1500, 1500 and 2000, respectively. These sizes resulted in the best clustering performance among the datasets. As expected, the smaller quantized data (particularly for size 100) resulted in a greater loss of information (because there is a risk of missing the clusters due to the small size) and therefore had a greater

detrimental effect on the GMM estimates. In S1, the value of the cluster means were not affected by a small value because of the well-separated clusters. Except for S1, the clusters were underestimated for a small size of n in the datasets S2, S3 and S4 where great discrepancies were observed in the cluster means because of the overlapping of clusters. The overlapped data merges the clusters where the performance of all the clustering algorithms is poor. For all sizes of n, the GMM performed better for S1 while for the other datasets, there was variation in the cluster means. Table 2 also shows that there was significant variation in the estimated cluster variances when using different sizes of the quantized data. Again, for small datasets, the cluster covariance matrices were different from the true covariance matrices of the clusters. However, as the size of the quantized data increased, the estimated covariance matrices became closer to the true covariance matrices of the clusters. Thus, the experimental results of the synthetic datasets show that if the size of the quantized data is not too small, the estimation of the cluster variances, in terms of quantized data, is performed accurately without applying the full data.

Table 2: The estimated clusters means and variances of different sizes of the quantized data along with the true ones.

Data set	Quantize data size	True Means	Cluster Means	True Variances	Cluster Variances
S1	100	**(-2.41, 2.36)**	(-2.39, 2.37)	**(1.35, 12.97)**	(1.86, 13.27)
	500		(-2.42, 2.38)		(1.57, 13.14)
	1000		**(-2.40, 2.36)**		**(1.40, 13.09)**
	1500		(-2.40, 2.38)		(1.40, 13.04)
	2000		(-2.40, 2.38)		(1.38, 13.03)
S2	100	**(8.17, 6.97, -3.25)**	(12.86, 2.38, NA)	**(4.91, 2.24, 2.23)**	(6.81, 8.29, NA)
	500		(10.01, 5.25, -4.99)		(5.14, 3.07, 2.99)
	1000		(8.90, 6.09, -3.91)		(4.62, 2.52, 2.49)
	1500		**(8.20, 6.73, -3.28)**		**(4.25, 2.31, 2.27)**
	2000		(8.21, 6.95, -3.27)		(4.20, 2.23,-3.28)
S3	100	**(-1.81, -6.38, -5.66, 1.92, -1.34)**	(-1.20, NA, -3.45, 4.98, NA)	**(4.57, 1.09, 3.53, 0.92, 5.69)**	(5.57, NA, 4.67, 3.97, NA)
	500		(-1.45, NA, -4.79, 2.44, -1.10)		(5.21, NA, 4.22, 1.86, 6.15)
	1000		(-1.85, -6.38, -5.66, 1.90, -1.33)		(4.59, 1.07, 3.53, 0.91, 5.68)
	1500		**(-1.80, -6.40, -5.70, 1.90, -1.34)**		**(4.57, 1.08, 3.55, 0.91, 5.68)**
	2000		(-1.81, -6.39, -5.66, 1.90, -1.33)		(4.58, 1.09, 3.53, 0.91, 5.68)
S4	100	**(0.81, -1.01, -5.90, 4.14, -0.32, -2.51, -0.61)**	(1.85, -1.79, -5.58, -6.42, -1.37, NA, NA)	**(1.14, 1.48, 2.18, 1.55, 2.26, 1.16, 5.14)**	(2.13, 4.67, 3.58, 1.23, 5.91, NA, NA)
	500		(0.80, -1.11, -5.95, 4.15, NA, NA, -0.56)		(1.25, 1.93, 2.39, 1.77, NA, NA, 5.18)
	1000		(0.62, NA, -5.86, 4.17, NA, -1.40, -0.61)		(1.29, NA, 2.31, 1.69, NA, 1.81, 5.07)
	1500		(0.68, -1.26, -5.87, 4.16, NA, NA, -0.60)		(1.15, 1.85, 2.30, 1.63, NA, NA, 4.98)
	2000		**(0.79, -1.05, -5.93, 4.14, -0.28, -2.56, -0.62)**		**(1.18, 1.50, 2.32, 1.62, 2.30, 1.12, 5.14)**

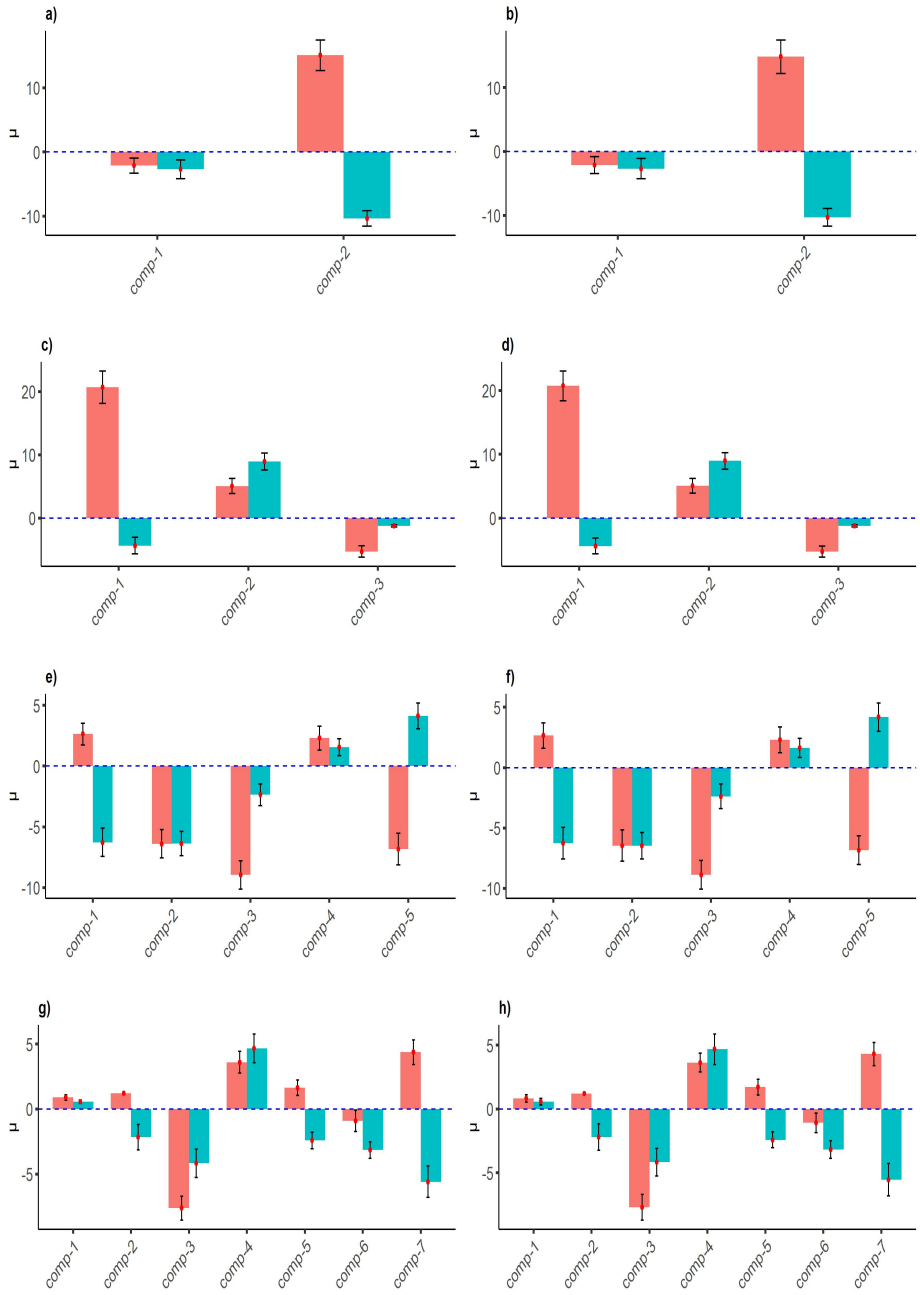

Fig. 3: (a) & (b) represent component means of the original and quantized dataset for S1, respectively. Similarly, (c) & (d) for S2, (e) & (f) for S3 and (g) & (h) for S4. The error bar is the 95% confidence interval on the means.

The mean vectors μ_k of the original and the quantized datasets are shown in Fig. 3, where a), c), e), and g) depicts the mean vectors of the full datasets S1, S2, S3, and S4, while b), d), f), and h) shows the mean vectors of the quantized datasets corresponding to their respective full datasets. Since every component has two means where orange bar shows the component mean for the x-axis whereas the metallic bar shows the component mean for the y-axis. Their output has the same statistical models with respect to their cluster centers and cluster structures. They reveal also that the estimates of the component means are similar in terms of accuracy, deduced from both the quantized values and the original datasets. A solid red dot on top of every bar shows the true mean of each component and the error bar shows the 95% confidence interval within which the true mean is to lie. The quantized dataset is smaller than the original dataset and it may be expected that the estimates based on the original datasets are more precise than the quantized values. In general, increasing the size of the quantized dataset resulted in a more accurate estimation of the parameter values.

Numerous techniques for model selection have been proposed for diverse settings [44]; for instance, Akaike's Information Criterion (AIC) [45], Bayesian Information Criterion (BIC) [40], sample-size-Adjusted AIC (AAIC) [46], Consistent AIC (CAIC) [47], and sample-size-Adjusted BIC (ABIC) [48]. Specific selections of A_n in Equation 6 make it equivalent to AIC, BIC, ABIC, or CAIC (see [44]). Among these techniques, the AIC and BIC stand out as extensively employed techniques in model selection [49]. Nylund et al. [50] conducted various sorts of simulations on the efficacy of different Information Criteria (IC) and tests for determining the number of classes in mixture models. Overall the simulation results showed that BIC performed better than other IC. The author recommended BIC as a common choice in various applications. The above-mentioned IC are neither right nor wrong. However, a researcher may evaluate that there could be a practical advantage to using one criterion over the other in certain cases. Sometimes, a preferred model using AIC may be too large, leading to challenges in usability and interpretation. For instance, in [51], BIC recommends the five-class mixture model, while AIC recommends a model with at least 10 classes. The authors concluded that interpreting a 10-class model would be hard to interpret; in such situations, the model favored by BIC is a notably better and more practical choice. However, the model selection and the number of clusters can be challenging. In this study, we prefer BIC for model selection and the number of clusters.

As described in section 3.1, the mclust package uses the BIC criterion to determine the optimal number of clusters by fitting the Gaussian Mixture Model with different numbers of clusters and comparing the BIC values of each model. The output of the quantized datasets is compared with the true model in terms of the geometric characteristics of the clusters. The geometric characteristics of the model are the volume, shape and orientation of the clusters. For instance, EVI denotes that all clusters share equal (E) volumes, variable (V) shapes and identity (I) orientation. The results are summarized in the Table 3. It can be observed that the small size of quantized data adequately describes the different properties of clusters such as the variability in volume, shape and orientation, but the variability in the number of clusters k and the size of the quantized dataset is inversely proportional. As the size of the quantized dataset increased, the variability in the number of clusters k

decreased. This indicates that using larger quantized datasets results in a more stable estimate of the number of clusters. The S2, S3 and S4 datasets have more complex structures and patterns compared to the S1 dataset, which leads to more variation in the geometrical characteristics of the clusters. As a result, these datasets required larger quantized datasets to achieve a stable estimate of the number of clusters and an accurate segmentation model. It is also observed that once the true model (VVV) was selected, it remained the same even after increasing the values of the quantized datasets.

Table 3: Model for different sizes of the quantized data along with the number of clusters.

	Quantize data size					
Datasets	100	500	1000	1500	2000	True Model
S1	VEV,2	VEV,2	**VVV,2**	VVV,2	VVV,2	**VVV,2**
S2	VVE,2	EVE,3	VEV,3	**VVV,3**	VVV,3	**VVV,3**
S3	VEV,3	VVE,5	VVE,5	**VVV,5**	VVV,5	**VVV,5**
S4	EEV,5	EVE,5	VVE,5	VVE,5	**VVV,7**	**VVV,7**

Additionally, the accuracy of the segmentation can be measured by the closeness of the estimated clusters to the true clusters. The selection of the number of components, k, is an important factor that affects the performance and accuracy of the segmentation. The number of clusters and optimal models of both the original and quantized datasets are depicted in Fig. 4. It is observed that both the datasets (original and quantized) had an identical number of clusters k in all the experiments. It is also observed if the size of quantized data was chosen appropriately, the loss of information did not affect the number of clusters obtained by the clustering algorithm. Thus, the quantized data allowed an efficient inference and an explicit representation of the underlying statistical structure which also performed well for the true parameter k.

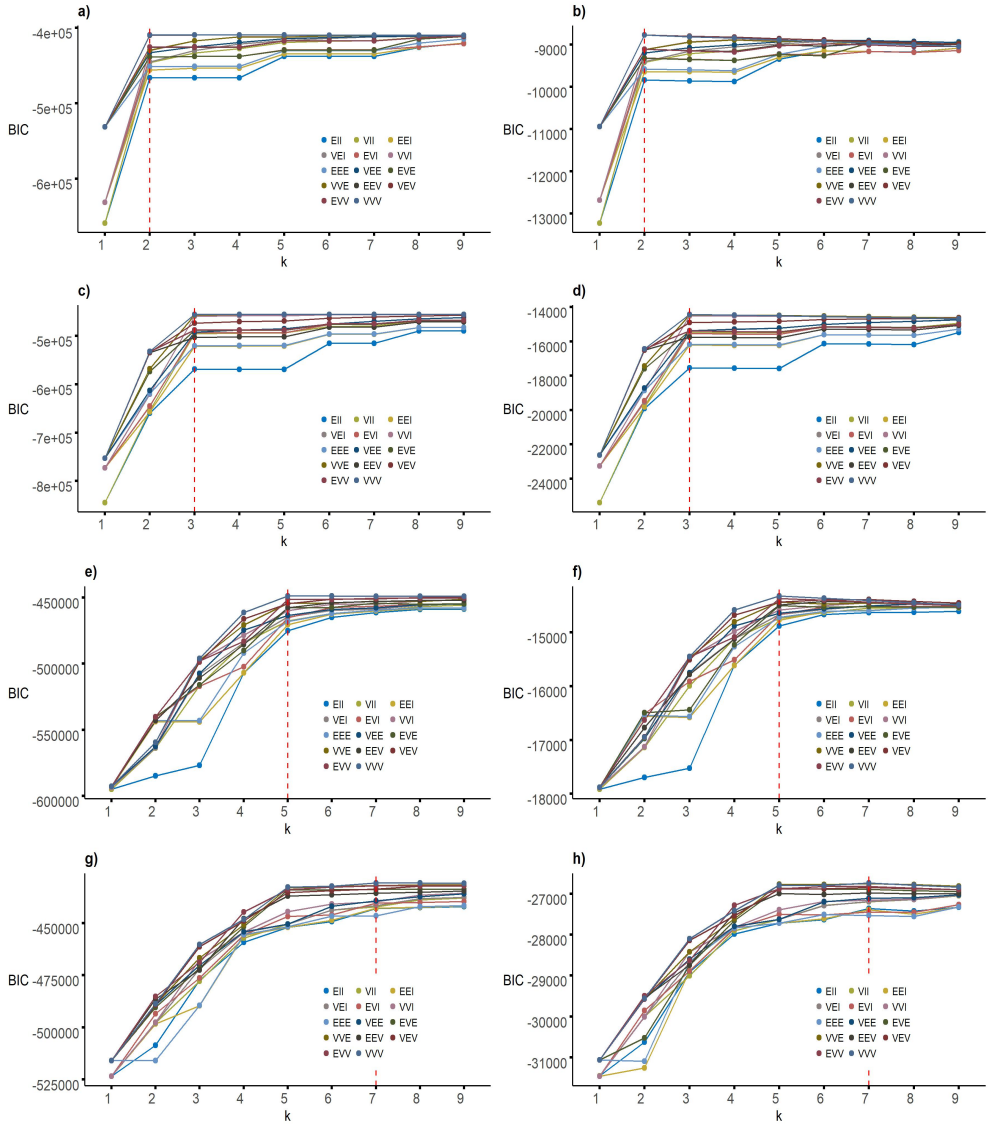

Fig. 4: (a) & (b) represent the number of clusters in the original and quantized datasets, respectively. Similarly, (c) & (d) for S2, (e) & (f) for S3 and (g) & (h) for S4. The legend shows the geometric characteristics of the model see section 3.2.

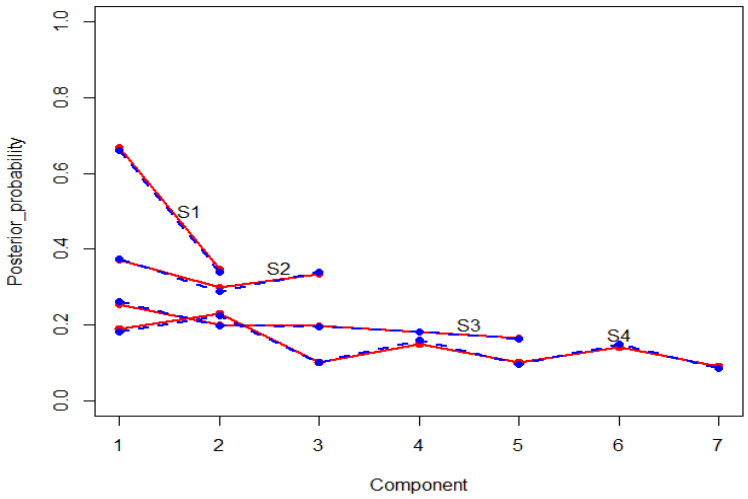

Fig. 5: Probability associated with each component of the original dataset (solid red line) vs the quantized data (blue dashed line) for datasets S1-S4.

The classification of the observations can be framed after allocating each of the objects in the group to which it has the highest estimated posterior probability. The performance of the quantized datasets was evaluated by comparing their posterior probabilities of the group's membership to that of the original datasets. The probability of the group membership of the original datasets and its respective quantized datasets are depicted in Fig. 5. It is observed that the estimated posterior probabilities of group membership are similar for both the original and quantized datasets. This suggests that the quantization process was successful in preserving the information of the original data and that the quantized data can be used as a good approximation of the original data. As a result of this observation, it is clear that quantized data will be preferred over original data for classification tasks due to the significant improvement in processing time.

4 Real-World Dataset

We have applied the proposed method on the four real-world datasets taken from the UCI machine learning repository [52]. These datasets have already well-specified known numbers of clusters; it is therefore, the researchers immensely use them for the assessment purposes. The basic characteristics of the real-world datasets are delineated in Table 4. One of the most important thing to be addressed here is to have a keen eye on the obtained results in regard of the real datasets because these datasets are mostly utilized in supervised learning techniques. It is therefore, we use them

rarely for solving the clustering related problems [53]. In order to evaluate the performance of the proposed algorithm, we apply the Adjusted Rand Index (ARI) [54] and Bayesian Information Criterion (BIC) [40]. ARI compares different clusters with the ground truth labels whereas BIC works as model selection at complex penalizing situation to fit the data accurately. The calculated score of ARI ranges from -1 to 1 of which 1 stands for the perfect similarity between the predicted and true labels. The highest BIC indicates a model that fits the data significantly. Moreover, the proposed algorithm is compared with the most common algorithms like K-means, PAM, CLARA, DBSCAN, CLARANS, Cure, Diana, and Agnes [55] by using different internal and external evaluation matrices. The internal measures consist of Dunn [56], Dunn2 [57] and Silhouette [58] whereas the external measures contained Purity, Entropy, and Normalized Mutual Information (NMI) [55]. The higher score of the internal and external indices indicates a better clustering performance, but entropy is an adverse measurement where the lower score means good clustering.

Table 4: The basic characteristics of real-world datasets.

Dataset	Number of Instances	Number of Attributes	Number of Clusters
Wine	178	13	3
Banknote Authentication	1372	4	2
Tripadvisor review	980	11	2
Vehicle	946	18	4

We conduct experiments on each data with diverse number of clusters k; 2, 3, 4 and 5. First of all, we select a subset from each real-world dataset for the reflection of its core properties. We apply the K-means clustering algorithm for the selection of a large number of small clusters which decreases the ratio of the variation within the clusters and the total variation. The scree plot of each real dataset as shown in Fig. 6 which depicts the ratio between the total sum of the square of within clusters and the total sum of the square of the clusters against the different values of k. For $k = 48$, the ratio reaches a local minimum and becomes constant for the wine data. Similarly, for $k = 240$, 280, and 150, the same ratio reaches the local minimum and becomes constant for the datasets banknote authentication, tripadvisor-review, and vehicle respectively.

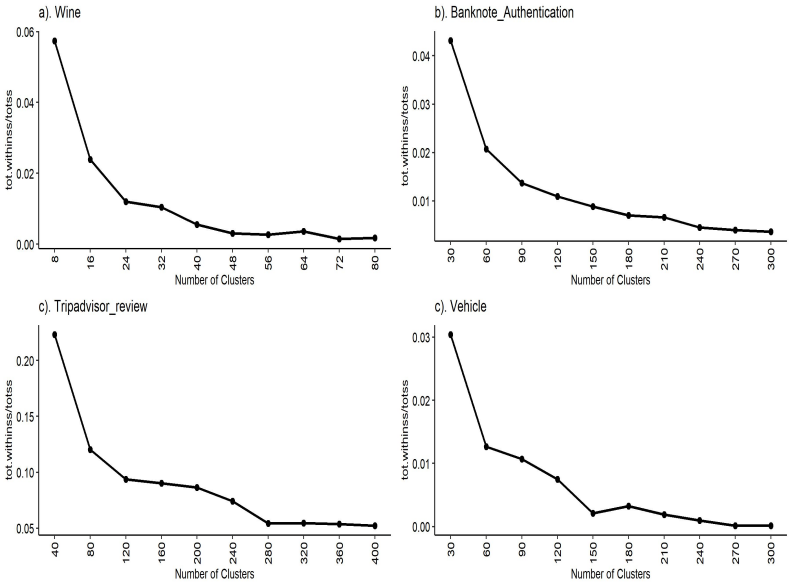

Fig. 6: The ratio of the within-cluster sum of the square to the total sum of the square after running K-means on each real-world data with different values of K.

The output of the traditional GMM and the proposed algorithm is summarized in Table 5. We have analyzed the wine data which consists of three clusters. As a result, the traditional algorithm achieves an ARI of 0.9667 while the proposed method exquisitely outperforms with an ARI of 0.9778 which indicates that the proposed method copes the clustering structure in a natural way which has great similarity with the true number of species. The proposed method has also a better BIC which keeps a more optimal balance between the model fit and complexity. However, the proposed method does not provide the BIC value for $k \geq 5$ which indicates that the proposed method may neglect the over-fitting. The banknote authentication dataset has two true clusters and both the proposed method and traditional GMM identify these two clusters correctly achieving the highest ARI and BIC scores. Similarly, we have analyzed the highest ARI for 2 clusters for the tripadvisor review dataset. However, the BIC score increases with the number of clusters which indicates that the traditional GMM is unsuccessful at pinpointing the optimal number of clusters. Contrary to this, the proposed method identifies the optimal number of clusters successfully with lower complexity. For the vehicle dataset, ARI improves as more as the number of clusters increases from 2 to 4 which shows that the proposed algorithm gets perfection for the identification of the true segments correctly. However, at 5 clusters, ARI drops slightly either of over-segmentation or splitting clusters unnecessarily. The results of both the metrics of the proposed algorithm portrays an efficient inference as well as the explicit representation of the underlying statistical structure which perform significantly for the true parameter k.

Table 5: The ARI and BIC values for the original dataset and the proposed method of the real-world datasets with 2, 3, 4, and 5 clusters.

		Original Dataset		**Proposed Method**	
Data Set	**K**	**ARI**	**BIC**	**ARI**	**BIC**
Wine	2	0.5820	-7026.543	0.6742	-3896.872
	3	**0.9667**	**-6849.387**	**0.9778**	**-3822.948**
	4	0.8174	-6885.515	0.96	-3852.597
	5	0.7358	-6956.809	0.805	NA
Banknote Authenticat	**2**	**0.9365**	**-23678.29**	**0.9534**	**-3742.948**
	3	0.8911	-25492.54	0.8263	-3832.271
	4	0.7876	-24906.22	0.8026	-3898.615
	5	0.7519	-24487.26	0.7609	-3797.590
Tripadvisor review	**2**	**0.8128**	2937.05	**0.8150**	**371.911**
	3	0.7819	3116.25	0.7327	317.124
	4	0.6517	**3382.45**	0.5845	273.326
	5	0.6257	3379.72	0.6331	245.900
Vehicle	2	0.55	11250.32	0.61	1278.426
	3	0.78	13480.45	0.75	1345.790
	4	**0.90**	**14175.65**	**0.91**	**1451.880**
	5	0.22	9367.09	0.20	850.907

Moreover, the proposed algorithm was compared with the most common algorithms such as K-means, PAM, Cure, CLARA, DBSCAN, CLARANS, Diana, and Agnes by using internal and external clustering validity indices. Table 6, displays the output of the cluster validation indices of each clustering algorithm on multiple datasets (wine, banknote authentication, tripadvisor review, and vehicle). It can be seen that the proposed algorithm achieves the highest purity among all the datasets, which indicates that it forms the most homogeneous clusters. Other methods like K-means, CLARA, PAM, Agnes, and CLARANS also perform relatively well but they are comparatively lower than the proposed method while DBSCAN, Cure, and Diana have the lowest purity which suggests poor separation between clusters. It is observed that the proposed algorithm has the lowest entropy followed by Diana and DBSCAN which indicates that it produces the most well-defined and less ambiguous clusters. The K-means, PAM, CLARANS, and Agnes have the same entropy score but they are comparatively higher than the proposed method where the Cure achieved the lowest entropy for wine dataset. Both the validity indices Dunn and Dunn2, have the highest value for the proposed algorithm which signifies well-defined clusters with minimal intra-cluster distance as followed by Cure, K-means and PAM, whereas the DBSCAN has the lowest index value for all the datasets and thus, it shown poor separation of clusters except the wine dataset.

Table 6: The output of different cluster validation indices of each clustering algorithm on multiple real datasets.

Datasets	Algorithm	Purity	Entropy	Dunn	Dunn2	Silhouette	NMI
Wine	K-means	0.7022	1.0863	0.1068	1.4067	0.5892	0.4287
	PAM	0.7079	1.0884	0.048	0.7492	0.5708	0.7515
	Cure	0.6461	0.5829	0.0923	1.0976	0.4337	0.3287
	CLARA	0.7191	1.0908	0.0162	1.293	0.5711	0.4342
	DBSCAN	0.4551	0.813	0.0968	0.1207	0.4097	0.0899
	CLARANS	0.7079	1.0891	0.0226	1.3614	0.5694	0.4173
	Diana	0.6573	0.8283	0.032	1.6075	0.5763	0.3491
	Agnes	0.6966	1.0849	0.0224	1.3684	0.5645	0.4159
	Proposed	0.9888	1.09	0.1402	1.6665	0.6723	0.9530
Banknote-authentication	K-means	0.6122	0.6388	0.0343	1.8136	0.4353	0.0292
	PAM	0.6217	0.691	0.0788	1.8865	0.4004	0.1044
	Cure	0.5554	0.693	0.0641	1.5972	0.4410	0.0035
	CLARA	0.6071	0.6918	0.0121	1.5859	0.4018	0.0359
	DBSCAN	0.5612	0.6716	0.0041	0.8327	0.3259	0.0119
	CLARANS	0.6232	0.6868	0.0072	1.653	0.4033	0.0411
	Diana	0.6108	0.6408	0.0145	1.8077	0.4315	0.0284
	Agnes	0.5554	0.693	0.0234	1.5972	0.3877	0.0035
	Proposed	0.6554	0.5793	0.0932	1.9536	0.5287	0.4048
Tripadvisor_review	K-means	0.8571	0.6687	0.0879	1.3684	0.3349	0.3498
	PAM	0.8704	0.6891	0.0648	1.8865	0.2838	0.2974
	Cure	0.7776	0.6623	0.1054	1.2877	0.2581	0.3009
	CLARA	0.85	0.6927	0.0758	1.2229	0.2547	0.2707
	DBSCAN	0.7918	0.4801	0.0645	0.9505	0.3166	0.1982
	CLARANS	0.8724	0.6931	0.0611	1.2571	0.267	0.36
	Diana	0.8551	0.6541	0.0691	1.2882	0.2953	0.2198
	Agnes	0.7776	0.6623	0.0785	1.2877	0.2793	0.2432
	Proposed	0.8776	0.3971	0.0967	1.9076	0.5613	0.4909
Vehicle	K-means	0.5532	1.3378	0.1447	1.2833	0.3385	0.2566
	PAM	0.5426	1.3366	0.0488	1.6865	0.4516	0.2515
	Cure	0.5426	1.3315	0.1361	1.2712	0.4848	0.2539
	CLARA	0.5426	1.3366	0.1147	1.2687	0.6617	0.2527
	DBSCAN	0.4149	1.2729	0.0810	1.6512	0.6411	0.1844
	CLARANS	0.5426	1.3366	0.0838	1.2687	0.4516	0.2527
	Diana	0.4362	1.1462	0.0904	0.6825	0.4376	0.1998
	Agnes	0.5426	1.3315	0.0838	1.2712	0.4499	0.2539
	Proposed	0.6319	1.0084	0.1791	1.8219	0.6088	0.6938

Similarly, the maximum Silhouette score of the proposed algorithm indicates well-clustered data with strong cohesion and separation as followed by K-means. The performance of other algorithms are same but comparatively lower than the proposed method. The CLARA algorithm performs well for wine and vehicle datasets but gives poor result for tripadvisor review and banknote authentication datasets. It is noted that the proposed algorithm has the highest NMI – Normalized Mutual Information than the other clustering algorithms which show a better agreement with ground truth labels. Fig. 7, depicts the comparative analysis of clustering algorithms using the real datasets across the various validity indices. The proposed algorithm performs more consistently than the other methods in terms of cluster quality, structure, and alignment with the ground truth. So, it is concluded that the proposed method has partitioned the real datasets correctly.

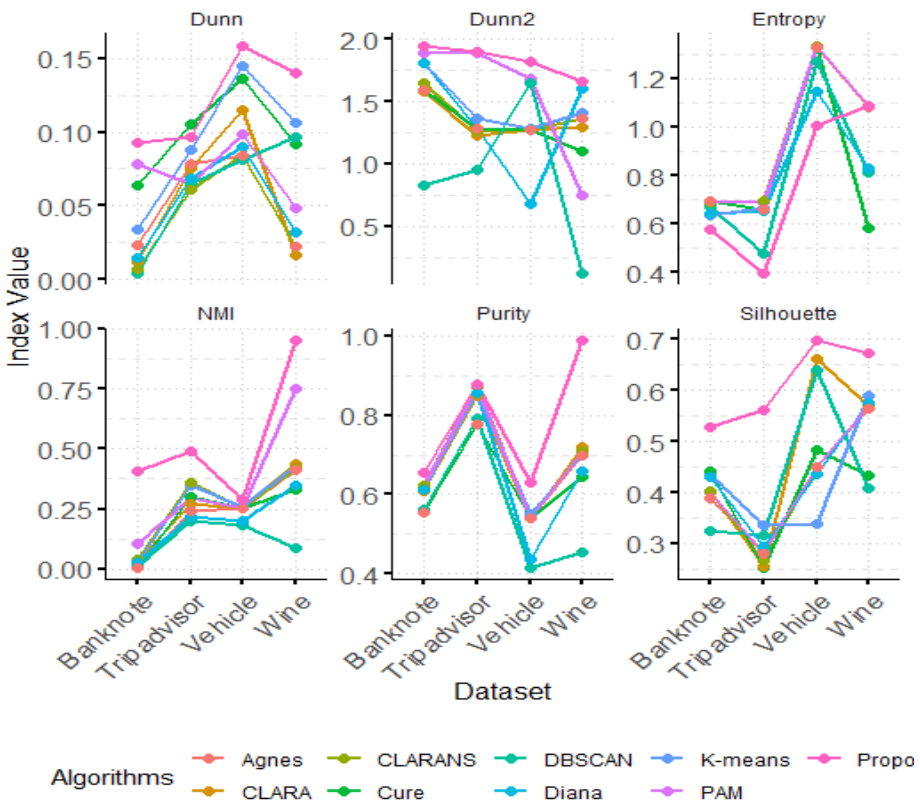

Fig. 7: The output of each validity index of different clustering algorithms for multiple real datasets.

5 Case Study

We applied the proposed method to analyze the satellite image of Islamabad, the capital city of Pakistan and its surrounding areas, as previously explored in the article [2], with the aim of determining the optimal number of clusters and to analyze and investigate the feature such as forest, urbanization, types of land use, etc. In this study, we used the 8 spectral bands with central wavelength, bandwidth and spatial resolution, respectively; these are displayed in Table 7. The spatial resolution for the bands 5-7 and 8A are 20 meters which were re-sampled to the 10-meter resolution. The acquired image is transformed into a usable data file by using the "raster" package in R [59]. The size of a data frame (an image) is around 1.63 gigabytes. Analyzing a dataset of this size requires substantial time and cost.

Table 7: Spectral properties of each band in the satellite image of Islamabad and its surroundings (June 2021).

Band Number	Central Wavelength (nm)	Bandwidth (nm)	Spatial Resolution (m)
2	490	65	10
3	560	35	10
4	665	30	10
5	705	15	20
6	740	15	20
7	783	20	20
8	842	115	10
8A	865	20	20

In the first stage of the proposed algorithm, we take a subset of data that keeps the salient features of the image. To accomplish this, we applied the K-means clustering algorithm to select a large number of small clusters, which effectively reduces the ratio of within-cluster variation to total variation. By doing so, the dataset becomes more computationally tractable for analysis. We explored a range of cluster numbers k, from 100 to 100,000. The importance of this wide range of cluster numbers is to determine the optimal size of the quantized dataset. The scree plot in Fig. 8 shows the ratio between the total sum of the square of the within clusters and the total sum of the square of the clusters for different values of k. It is clear from the figure that a small number of clusters cannot represent properly the whole dataset. For $k = 50{,}000$, the ratio reaches a local minimum and becomes constant. Hence, 50,000 cluster centers are used instead of the 1.63 gigabytes dataset in the proposed method. This strategic choice not only streamlines the computational process but also preserves the essential characteristics of the data for our analysis.

In the second stage of the algorithm, we utilized the reduced dataset in a GMM analysis to determine the final number of clusters to classify our image. Fig. 9 shows

the gradient of the BIC score, which suggests 10 and 11 clusters (however, we will be using 10 clusters). We also fit a GMM to the full image dataset. Using the original image data, 8 clusters are selected as depicted in Fig. 10. The disparity in the number of clusters between the reduced and original datasets does not exert any discernible impact on the clustering outcomes, as shown in Fig. 10, where clusters 8, 9 and 10 all yield the same gradient scores.

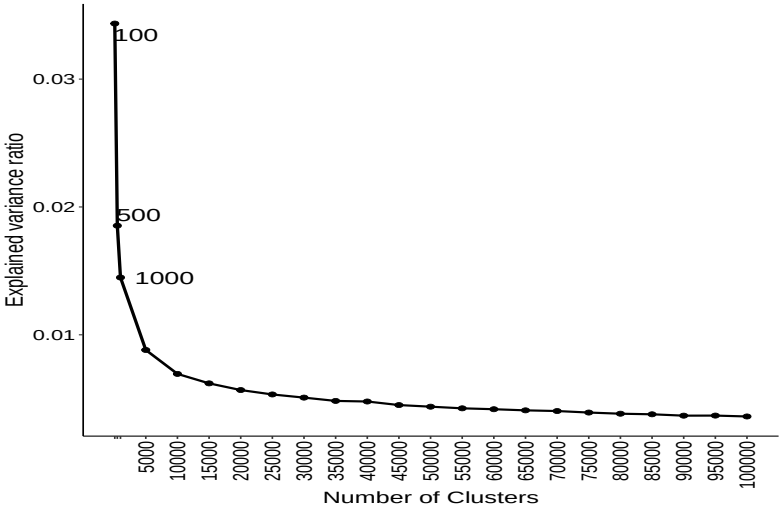

Fig. 8: The ratio of the within-cluster sum of the square to the total sum of the square after running K-means for different values of K.

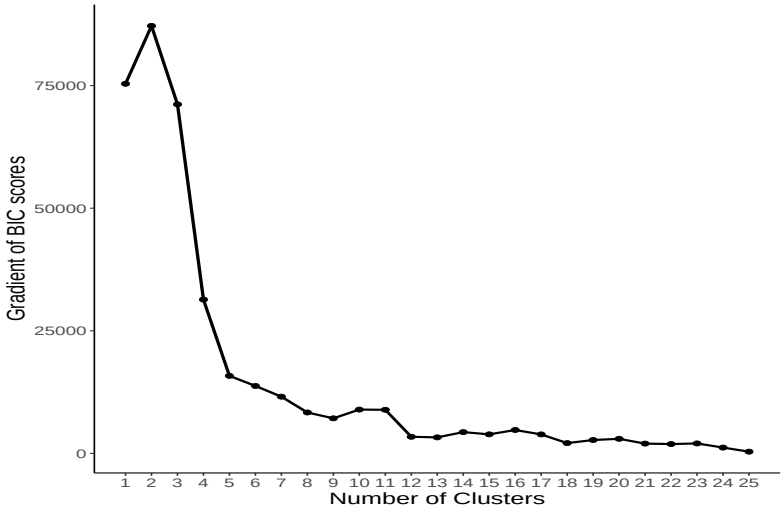

Fig. 9: Plot of the gradient of Bayesian Information Criterion (BIC) of Gaussian Mixture Model from 1 to 25 components using the reduced quantized dataset of size 50,000.

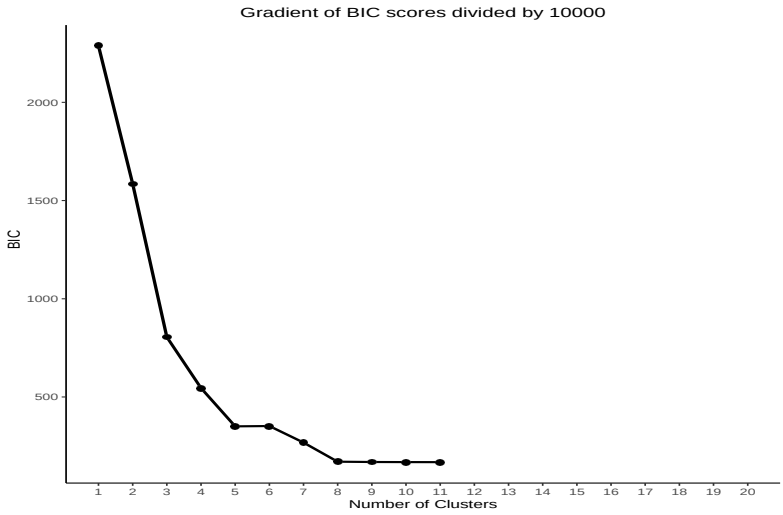

Fig. 10: Plot of the gradient of Bayesian Information Criterion (BIC) of Gaussian Mixture Model from 1 to 25 components using the full dataset of size of around 1.63 gigabytes.

To classify the satellite image into different geographical areas such as the forest, urban and other areas, the Gaussian Mixture Model is applied to both the reduced (50,000 K-means cluster centers) and the original datasets. Fig. 11 shows the snapshot of the 2021 satellite image which bears the 10 numbers of clusters of the same features. The largest cluster, which is grey in color, shows the agricultural area. The pink cluster, on the other hand, indicates agricultural areas of green crops. The dark-green cluster represents the thick forest region specifically the Margalla Hills. The second and seventh clusters depict the areas of sparse forest, encompassing both the mountainous terrain and planned areas such as Lake View, F-9 and Ayub parks. The yellow cluster represents the solar panels and the roofs of the urban area. The red cluster represents the urbanization including the densely populated commercial and residential areas. The cluster number 8 represents both the rugged surfaces and the landscaped land. The black color cluster indicates the pockets of greenery situated on the sides of roads and rivers and the number of trees in the houses respectively. The blue cluster (small in size) describes a body of water (including the Rawal Lake, the Khanpur Dam downstream, the Korang River etc).

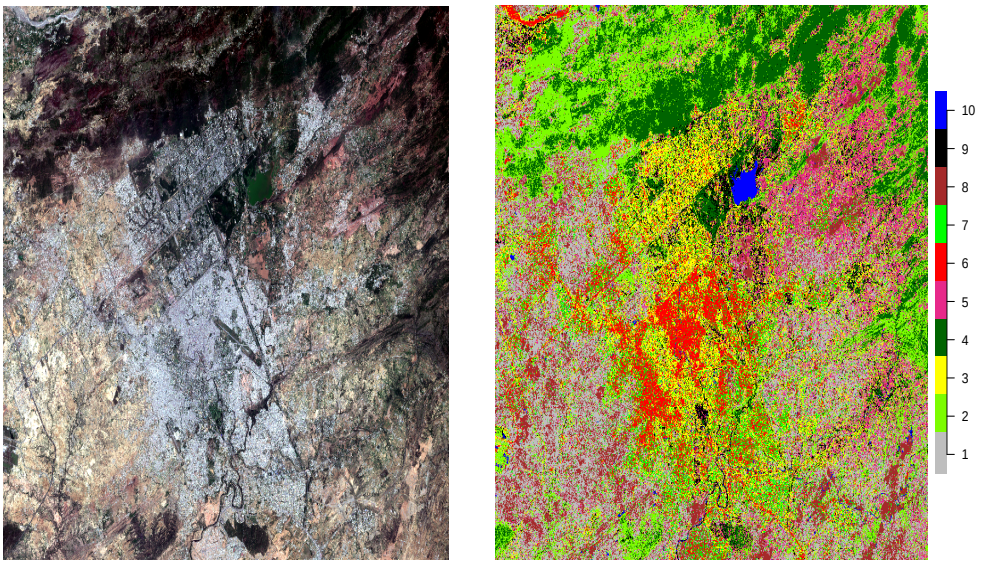

Fig. 11: The original satellite image (June 2021) of Islamabad and its surroundings and its clustered image with a total of ten clusters using our proposed method.

The cluster of the original image of June 2021 is shown in Fig. 12. The image is clustered into the 8 numbers of clusters by using the method as depicted in Fig. 10. In the cluster of the original image, the white cluster consists of buildings with dark

grey roofs (old residential areas and solar panels) and land that has not been harvested (barren land). The yellow cluster indicates the bushes and grass on roadsides and the river banks as well as agricultural areas of green crops. The red cluster represents urbanization with a mix of both commercial and high-density residential areas. The fourth and seventh clusters represent the forest regions of both the mountainous terrains and planned areas. The grey cluster indicates the agricultural land as well as the planned area. The brown color encompasses both the rocky and landscaped land. The blue color cluster represents some particular bodies of water including the Rawal Lake, the downstream of Khanpur Dam, the Korang River, etc. The difference in the number of clusters between the reduced and original datasets does not exert any influence on the clustering output. Thus, the proposed algorithm not only demonstrates a better ability for accurate segmentation but also performs better in terms of speed and computational efficiency. The GMM method took one week to analyze the original dataset consisting of 1.63 gigabytes, while the proposed algorithm reduced the computational time from week to hours, using Google Cloud with 32 GB RAM.

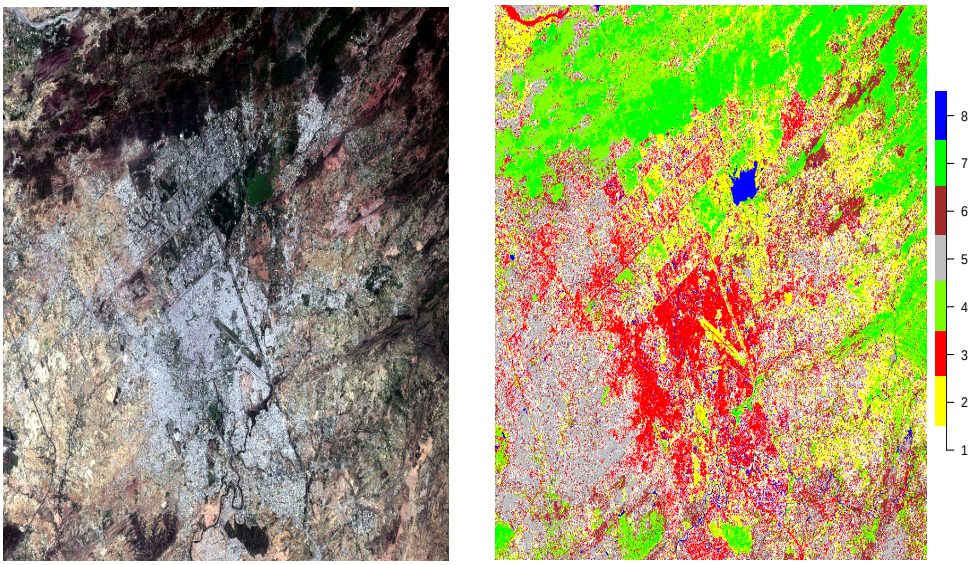

Fig. 12: The original satellite image (June 2021) of Islamabad and its surroundings and its clustered image with a total of eight clusters using the traditional Gaussian Mixture Model.

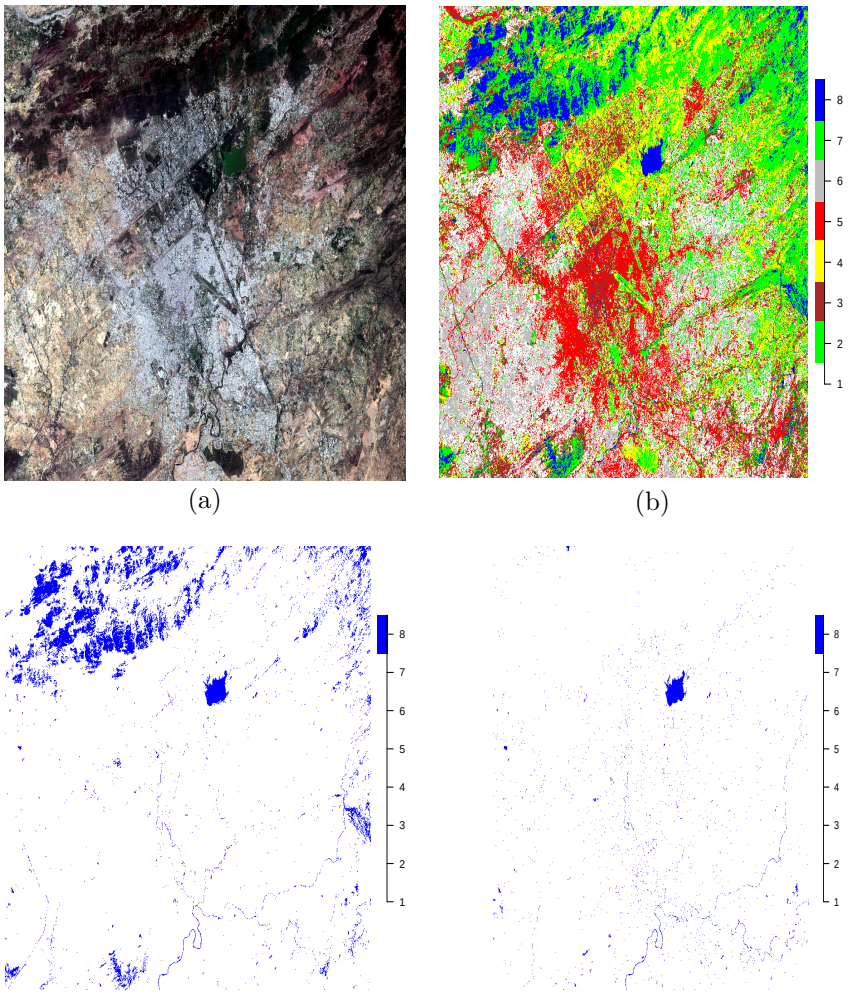

Fig. 13: (a) Original satellite imagery, (b) clustered image using the method proposed by Ali et al., (2022) [2], (c) water bodies extracted from (b), and (d) water bodies extracted from Fig. 12.

We compared the Ali et al., [2] method with the newly proposed Gaussian Mixture Model to evaluate their performance on satellite images for the correct identification of the water bodies. As shown in Fig. 13, subfigure (c) shows that the previous method often confused water bodies with forest areas, merging them into a single cluster and causing misclassification. In contrast, our proposed method, shown in subfigure (d), clearly separates water bodies from forest areas, resulting in a distinct classification.

This improvement is significant for applications such as environmental monitoring and water resource management. These results demonstrate that our approach is more accurate than the previous method and more effective for land cover analysis in satellite-based studies.

6 Conclusion

The size of a dataset has an immense effect on the time, memory efficiency and quality of the cluster. Most of the existing clustering methods perform well when the data are small in size. For big data, conventional methods often become impractical due to their limitations in efficiency and scalability as they need the development of faster and more insightful approaches. Therefore, a balance must be struck between computational efficiency and accuracy when choosing a clustering method for big data. The Gaussian Mixture Model is widely used as an accurate and informative data-driven way to identify clusters in the data. However, for large datasets, the Gaussian Mixture Model is computationally expensive.

The aim of this study is the scaling of GMM to large datasets such as Landsat imagery. We achieved this task using the widely adopted K-means clustering algorithm, known for its scalability and efficiency for big data to extract quantized data. Data quantization can be useful in reducing the size of the dataset and removing redundant information which in turn reduces the computational burden of the clustering algorithm. The K-means clustering is computationally faster and less expensive while the Gaussian Mixture Model accounts for the correlation between the variables and also computes the number of components in a data-driven manner.

The simulation results showed that a balanced number of quantized values provide good clustering results that closely match the actual partition in the underlying distribution. The correlation between the quantized data and original data exceeds 0.98 in all cases. The smaller quantized values lead to underestimation of the number of clusters and the clustering structure because there is a risk of missing small clusters. The variability in the number of clusters and the size of the quantized dataset is inversely proportional. The simulation study shows that the quantized dataset represents the original datasets well and can be used as a proxy for the larger datasets. The proposed algorithm not only reduces the computational burden but also reduces the impact of noise on the clustering results, making it a valuable tool in big data clustering.

Moreover, we extended our proposed algorithm to the satellite images of Islamabad region, the capital city of Pakistan and its environs, with the aim of determining the optimal number of clusters, to analyze and investigate features such as forest, urbanization, types of land use, etc. A notable capability of the proposed algorithm is its ability to describe the various water bodies within the image which the traditional K-means clustering algorithm fails to detect (for example, see [2]). The proposed algorithm not only demonstrates a better ability for accurate segmentation but also performs better in terms of speed and computational efficiency. The original image, consisting of 1.63 gigabytes of data, required an entire week to analyze. The proposed algorithm reduces the computational burden from weeks to hours.

The quantization process can be enhanced by incorporating clusters spread along with the cluster centers. We leave the exploration of these extensions for future research endeavors.

Abbreviations. GMM: Gaussian Mixture Model; EM: Expectation Maximization; GPU: Graphics Processing Units; CLARANS: Clustering Algorithm based on Randomized Search; CLARA: Clustering for Large Applications; MCMC: Markov Chain Monte Carlo; WCSS: within-cluster sum of squares; TCSS: total cluster sum of squares; AIC: Akaike's Information Criterion; BIC: Bayesian Information Criterion; ABIC: sample-size-Adjusted BIC; CAIC: Consistent AIC; ABIC: sample-size-Adjusted BIC.

Acknowledgments. This research was conducted while the first author was hosted by the QUT Center for Data Science.

Authors' contributions. Conceptualization, IA, AUR, KM and DMK; methodology, IA, KM; software, IA and AUR; validation, IA; formal analysis, IA and AUR; investigation, IA; resources, AUR, KM and DMK; data curation, IA and AUR; writing–First draft preparation, IA; writing–review and editing, KM and DMK; visualization, IA, AUR, K.M and DMK; supervision, KM and DMK; project administration, KM and DMK.

Conflict of interest. The authors declare that they have no competing interests.

Ethics approval and Consent to participate. Not applicable

Consent for publication. All authors read the final manuscript and approved it for publication.

Funding. The first author received the financial support from the Higher Education Commission of Pakistan and the Center for Data Science, Queensland University of Technology, Australia.

Availability of data and materials. The datasets utilized and/or analyzed in the present study can be obtained from the corresponding author upon a reasonable request.

Appendix A EM Algorithm

Let z_i be the latent variable denoting the unknown membership label of the observed data x_i. Then, the complete data can be expressed as $\mathcal{Y} = \{x, z\}$. The log-likelihood function of the complete dataset is then given by:

$$log(\ell(\psi \mid \mathcal{Y})) = \sum_{i=1}^{n} \sum_{k=1}^{M} z_{ik} \log\left(\langle \pi_k \mathcal{N}(x_i \mid \psi_k) \rangle\right) \tag{A1}$$

where z_{ik} is an indicator variable that takes the value 1 if observation x_i belongs to the k^{th} component and 0 otherwise. The EM algorithm consists of an expectation step (E-step) and a maximization step (M-step). In the (E-step), the objective is to

determine the conditional expectation of the complete-data log-likelihood function based on the observed data. This expectation is commonly known as the $\mathcal{Q}$-function and can be defined as follow:

$$\mathcal{Q}(\psi, \hat{\psi}^{(t)}) = \mathbb{E}_{\hat{\psi}^{(t)}}[log(\ell(\psi \mid \mathcal{Y}))] = \sum_{i=1}^{n}\sum_{k=1}^{M} \hat{\gamma}_{ik}^{t} \langle \log \pi_k + \log \mathcal{N}(x_i \mid \psi_k) \rangle \tag{A2}$$

where $\hat{\gamma}_{ik}^{t} = \mathbb{E}_X \left[z_i = k \mid x, \psi_k^{(t-1)} \right]$ is the posterior probability $\mathcal{P}\left[z_i = k \mid x, \psi_k^{(t-1)} \right]$, which can be interpreted as the probability of the i^{th} observation being generated by the k^{th} Gaussian component under the current set of parameters, where t represents the number of iterations. The prior probability of a particular observation x being assigned to cluster k is $\mathcal{P}(z_i = k \mid x) = \pi_k$. The posterior probabilities $\hat{\gamma}_{ik}^{t}$, $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, M$, can be expressed using Bayes' theorem as follows:

$$\begin{aligned} \mathcal{P}(z_i = k \mid x, \psi^{(t-1)}) &= \frac{\mathcal{P}(x, z_i)}{\mathcal{P}(x)} \\ &= \frac{\mathcal{P}(z_i = k)\mathcal{P}(x \mid z_i)}{\sum_{k=1}^{M} \mathcal{P}(z_k)\mathcal{P}(x \mid z_k)} \\ &= \frac{\hat{\pi}_i^{(t-1)} \mathcal{N}(x \mid \psi_i^{(t-1)})}{\sum_{k=1}^{M} \hat{\pi}_k^{(t-1)} \mathcal{N}(x \mid \psi_k^{(t-1)})} = \hat{\gamma}_{ik}^{(t)} \end{aligned} \tag{A3}$$

The M-step maximizes the conditional expectation of the complete log-likelihood function concerning the observed data x given the current parameters i.e, $\psi^{t+1} = argmax\mathcal{Q}(\psi, \hat{\psi(t)})$. In case of GMM the maximization of $\mathcal{Q}(\psi, \hat{\psi(t)})$ involves update of the means $\mu^{(t)}$, the covariance matrices $\Sigma^{(t)}$ and mixture proportions $\pi^{(t)}$ by the following equations:

$$\hat{\mu}_k^{(t)} = \frac{\sum_{i=1}^{n} \hat{\gamma}_{ik}^{(t)} x_i}{\sum_{i=1}^{n} \hat{\gamma}_{ik}^{(t)}} = \frac{1}{N_k^{(t)}} \sum_{i=1}^{n} \hat{\gamma}_{ik}^{(t)} x_i \,, \tag{A4}$$

$$\hat{\Sigma}_k^{(t)} = \frac{1}{N_k^{(t)}} \sum_{i=1}^{n} \hat{\gamma}_{ik}^{(t)} (x_i - \mu_k^{(t)})(x_i - \mu_k^{(t)})^T, \tag{A5}$$

and

$$\hat{\pi}_k^{(t)} = \frac{N_k^{(t)}}{n}. \tag{A6}$$

where ${N_k}^{(t)} = \sum_{i=1}^{n} \hat{\gamma}_{ik}^{(t)}$
Generally, the EM algorithm follows the following four steps to estimate the parameters ψ.

1. First initialize the parameters μ_k, Σ_k and π_k and evaluate the log-likelihood function with these parameters.
2. Expectation (E-step): The E-step evaluates the posterior probabilities $\hat{\gamma}_{ik}$ under the current set of the parameters $\psi = \{\pi_k, \mu_k, \Sigma_k\}$ with equation A3.
3. Maximization (M-step): The maximization step re-estimates the parameters $\hat{\mu}_k$, $\hat{\Sigma}_k$ and $\hat{\pi}_k$ using the current values of $\hat{\gamma}_{ik}$ in equations A4, A5 and A6.
4. If the difference between two iterations of the log-likelihood function is less than some small ϵ (threshold), the process is converged. Otherwise, repeat step 2.

For further details of the EM algorithm, interested readers are referred to [4].

References

[1] Yin, S., Zhang, Y., Karim, S.: Large scale remote sensing image segmentation based on fuzzy region competition and gaussian mixture model. IEEE Access **6**, 26069–26080 (2018) https://doi.org/10.1109/ACCESS.2018.2834960

[2] Ali, I., Rehman, A.U., Khan, D.M., Khan, Z., Shafiq, M., Choi, J.-G.: Model selection using k-means clustering algorithm for the symmetrical segmentation of remote sensing datasets. Symmetry **14**(6), 1149 (2022)

[3] Patel, E., Kushwaha, D.S.: Clustering cloud workloads: K-means vs gaussian mixture model. Procedia computer science **171**, 158–167 (2020)

[4] Bishop, C.: Pattern recognition and machine learning. Springer google schola **2**, 35–42 (2006)

[5] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological) **39**(1), 1–22 (1977)

[6] Kurban, H., Jenne, M., Dalkilic, M.M.: Using data to build a better em: Em* for big data. International Journal of Data Science and Analytics **4**(2), 83–97 (2017)

[7] Panić, B., Klemenc, J., Nagode, M.: Improved initialization of the em algorithm for mixture model parameter estimation. Mathematics **8**(3), 373 (2020)

[8] Chang, J., Fisher III, J.W.: Parallel sampling of dp mixture models using sub-clusters splits. In: Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 1, pp. 620–628 (2013)

[9] Williamson, S., Dubey, A., Xing, E.: Parallel markov chain monte carlo for non-parametric mixture models. In: International Conference on Machine Learning (ICML-13), pp. 98–106 (2013)

[10] Lee, A., Yau, C., Giles, M.B., Doucet, A., Holmes, C.C.: On the utility of graphics cards to perform massively parallel simulation of advanced monte carlo methods. Journal of computational and graphical statistics **19**(4), 769–789 (2010)

[11] Ordonez, C., Omiecinski, E.: Frem: fast and robust em clustering for large data sets. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 590–599 (2002)

[12] Verma, N.K., Dwivedi, S., Sevakula, R.K.: Expectation maximization algorithm made fast for large scale data. In: 2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI), pp. 1–7 (2015). IEEE

[13] Neagoe, V.-E., Chirila-Berbentea, V.: Improved gaussian mixture model with

expectation-maximization for clustering of remote sensing imagery. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 3063–3065 (2016). IEEE

[14] Ormerod, J.T., Wand, M.P.: Explaining variational approximations. The American Statistician **64**(2), 140–153 (2010)

[15] Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. Journal of the American statistical Association **112**(518), 859–877 (2017)

[16] Marin, J.-M., Pudlo, P., Robert, C.P., Ryder, R.J.: Approximate bayesian computational methods. Statistics and Computing **22**(6), 1167–1180 (2012)

[17] Moores, M.T., Drovandi, C.C., Mengersen, K., Robert, C.P.: Pre-processing for approximate bayesian computation in image analysis. Statistics and Computing **25**(1), 23–33 (2015)

[18] Guha, S., Rastogi, R., Shim, K.: Cure: An efficient clustering algorithm for large databases. ACM Sigmod record **27**(2), 73–84 (1998)

[19] Ng, R.T., Han, J.: Clarans: A method for clustering objects for spatial data mining. IEEE transactions on knowledge and data engineering **14**(5), 1003–1016 (2002)

[20] Kaufman, L.: Clustering large data sets. Pattern recognition in practice, 425–437 (1986)

[21] Saeed, M.M., Al Aghbari, Z., Alsharidah, M.: Big data clustering techniques based on spark: a literature review. PeerJ Computer Science **6**, 321 (2020)

[22] Huang, Z., Gelman, A.: Sampling for bayesian computation with large datasets. Available at SSRN 1010107 (2005)

[23] Manolopoulou, I., Chan, C., West, M.: Selection sampling from large data sets for targeted inference in mixture modeling. Bayesian analysis (Online) **5**(3), 1 (2010)

[24] Diday, E.: Thinking by classes in data science: the symbolic data analysis paradigm. Wiley Interdisciplinary Reviews: Computational Statistics **8**(5), 172–205 (2016)

[25] Beranger, B., Lin, H., Sisson, S.: New models for symbolic data analysis. Advances in Data Analysis and Classification **17**(3), 659–699 (2023)

[26] Shaham, U., Stanton, K., Li, H., Nadler, B., Basri, R., Kluger, Y.: Spectralnet: Spectral clustering using deep neural networks. arXiv preprint arXiv:1801.01587 (2018)

[27] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)

[28] Zhu, X.X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F.: Deep learning in remote sensing: A comprehensive review and list of resources. IEEE geoscience and remote sensing magazine **5**(4), 8–36 (2017)

[29] Goodfellow, I.: Deep learning. MIT press (2016)

[30] Stamoulis, D., Cai, E., Juan, D.-C., Marculescu, D.: Hyperpower: Power-and memory-constrained hyper-parameter optimization for neural networks. In: 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 19–24 (2018). IEEE

[31] Gong, P., Yu, L., Li, C., Wang, J., Liang, L., Li, X., Ji, L., Bai, Y., Cheng, Y., Zhu, Z.: A new research paradigm for global land cover mapping. Annals of GIS **22**(2), 87–102 (2016)

[32] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[33] Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015)

[34] Buhrmester, V., Münch, D., Arens, M.: Analysis of explainers of black box deep neural networks for computer vision: A survey. Machine Learning and Knowledge Extraction **3**(4), 966–989 (2021)

[35] Ullah, I., Mengersen, K.: Bayesian mixture models and their big data implementations with application to invasive species presence-only data. Journal of Big Data **6**(1), 1–25 (2019)

[36] De Vries, C.M., De Vine, L., Geva, S., Nayak, R.: Parallel streaming signature em-tree: A clustering algorithm for web scale applications. In: Proceedings of the 24th International Conference on World Wide Web, pp. 216–226 (2015)

[37] MacQueen, J., *et al.*: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967). Oakland, CA, USA

[38] Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics) **28**(1), 100–108 (1979)

[39] Borchers, H.W.: Pracma: Practical Numerical Math Functions. (2022). R package

version 2.3.8. https://CRAN.R-project.org/package=pracma

[40] Schwarz, G.: Estimating the dimension of a model. The annals of statistics, 461–464 (1978)

[41] Milligan, G.W., Isaac, P.D.: The validation of four ultrametric clustering algorithms. Pattern Recognition **12**(2), 41–50 (1980)

[42] Fränti, P., Sieranoja, S.: K-means properties on six clustering benchmark datasets. Applied Intelligence **48**(12), 4743–4759 (2018)

[43] Fraley, C., Raftery, A., Scrucca, L., Murphy, T., Fop, M.: Package "mclust": Gaussian Mixture Modelling for Model-Based Clustering, Classification, and Density Estimation. The Comprehensive R Archive Network (2016)

[44] Dziak, J.J., Coffman, D.L., Lanza, S.T., Li, R., Jermiin, L.S.: Sensitivity and specificity of information criteria. Briefings in bioinformatics **21**(2), 553–565 (2020)

[45] Akaike, H.: Information theory and an extension of the maximum likelihood principle. In: Selected Papers of Hirotugu Akaike, pp. 199–213. Springer, ??? (1998)

[46] Hurvich, C.M., Tsai, C.-L.: Regression and time series model selection in small samples. Biometrika **76**(2), 297–307 (1989)

[47] Bozdogan, H.: Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. Psychometrika **52**(3), 345–370 (1987)

[48] Sclove, S.L.: Application of model-selection criteria to some problems in multivariate analysis. Psychometrika **52**, 333–343 (1987)

[49] Liu, Q., Charleston, M.A., Richards, S.A., Holland, B.R.: Performance of akaike information criterion and bayesian information criterion in selecting partition models and mixture models. Systematic Biology **72**(1), 92–105 (2023)

[50] Nylund, K.L., Asparouhov, T., Muthén, B.O.: Deciding on the number of classes in latent class analysis and growth mixture modeling: A monte carlo simulation study. Structural equation modeling: A multidisciplinary Journal **14**(4), 535–569 (2007)

[51] Chan, W.-H., Leu, Y.-C., Chen, C.-M.: Exploring group-wise conceptual deficiencies of fractions for fifth and sixth graders in taiwan. The Journal of Experimental Educational, 26–57 (2007)

[52] Dua, D., Graff, C.: UCI Machine Learning Repository (2017). http://archive.ics.uci.edu/ml

[53] Guyon, I., Von Luxburg, U., Williamson, R.C.: Clustering: Science or art. In: NIPS 2009 Workshop on Clustering Theory, pp. 1–11 (2009)

[54] Hubert, L., Arabie, P.: Comparing partitions. Journal of classification **2**, 193–218 (1985)

[55] Anand, S.K., Kumar, S.: Experimental comparisons of clustering approaches for data representation. ACM Computing Surveys (CSUR) **55**(3), 1–33 (2022)

[56] Dunn, J.C.: Well-separated clusters and optimal fuzzy partitions. Journal of cybernetics **4**(1), 95–104 (1974)

[57] Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Journal of intelligent information systems **17**, 107–145 (2001)

[58] Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)

[59] Hijmans, R., Van Etten, J., Cheng, J., Mattiuzzi, M., Sumner, M., Greenberg, J., Lamigueiro, O., Bevan, A., Racine, E., Shortridge, A., et al.: Package 'raster'. R Package. (accessed 1 October 2016). https://cran.r-project.org/web/packages/raster/index.html