

A Scalable Clustering-Based Approach for Efficient Mining of Large Datasets

Harshita Koushal¹, Dr. Harsh Lohiya², Mr. Manoj Kumar Yadav³

¹Research Scholar, ²Associate Professor, ³Assistant Professor,

Department of CSE

Sri Satya Sai University of Technology & Medical Science (SSSUTMS), Sehore, Bhopal, Madhya Pradesh

Abstract: Clustering is an unsupervised machine learning technique used to identify and group similar data points within large datasets without relying on predefined class labels or target outcomes. It plays an important role in data mining by facilitating the discovery of meaningful groups, hidden patterns, and relevant data distributions. Traditional clustering methods generally assume that clusters are spherical and have relatively similar sizes, which can reduce their effectiveness when dealing with irregularly shaped clusters or datasets containing outliers. To address the challenges associated with clustering large databases, several approaches employ random sampling and partitioning techniques. In such methods, a large dataset is first divided into smaller partitions, and each partition is clustered independently. The resulting partial clusters are subsequently combined and clustered again to generate the final set of clusters. This approach reduces the computational burden associated with processing the entire dataset simultaneously. With the rapid growth of data volumes, several parallel clustering techniques based on the Map Reduce framework have also been proposed to improve scalability. However, applying the conventional K-Means algorithm to large datasets in a parallel environment may require the repeated reading and processing of data during each iteration. Consequently, this iterative process can significantly increase input/output operations and network communication overhead. To overcome these limitations, a series-based K-Means clustering approach, referred to as CBK Means, can be employed to reduce the amount of data processed while maintaining or improving clustering quality. The method uses representative data verification to effectively reduce the dataset size before or during the clustering process. Experimental results indicate that CBK Means provides improved efficiency and scalability compared with the conventional K-Means algorithm, while also achieving better clustering accuracy.

Keywords: Data mining, knowledge discovery, clustering algorithms, sampling

I. INTRODUCTION

Advances in science and technology have significantly simplified the collection, storage, and management of large volumes of data while also reducing the associated costs. However, extracting meaningful and useful information from massive datasets generated in scientific, governmental, and industrial domains requires efficient data processing and analysis techniques. A common example can be observed in search engines, where a user may receive thousands or even millions of results for a single query. Manually examining such a large number of results to identify relevant information can be time-consuming and inefficient. Clustering can address this problem by grouping similar search results into meaningful categories, enabling users to focus on relevant groups while avoiding unrelated documents. The fundamental objective of clustering is to identify the underlying structure of a dataset and organize its data objects into distinct groups, known as clusters, such that objects within the same cluster exhibit a high degree of similarity while objects belonging to different clusters are comparatively dissimilar.



Applications of Clustering

Clustering techniques have been successfully applied in numerous domains. Some important applications include:

- **News Article Categorization:** Clustering can be used to automatically organize daily news articles into meaningful categories such as sports, entertainment, business, politics, technology, and health.
- **Web Document Classification:** Search engine results can be grouped according to their similarity to a user's query. This enables related web pages or documents to be presented together, making information retrieval more efficient.
- **Market Analysis:** Clustering can be applied to large databases containing customers' purchasing histories to identify groups of consumers with similar purchasing patterns, preferences, or behavioral characteristics. Such information can support customer segmentation and targeted marketing strategies.
- **Scientific Research:** Modern research activities generate enormous quantities of data through sensors, experiments, simulations, and other data-collection systems. Clustering helps researchers identify significant patterns, relationships, and correlations within these datasets, thereby supporting meaningful interpretations and discoveries.
- **Earthquake Analysis:** Clustering can be applied to earthquake data by grouping observed epicenters according to their geographical and spatial characteristics. This can assist researchers in identifying regions that may have a higher risk of seismic activity.

Scalable Clustering for Large Datasets

Conventional clustering techniques, such as K-Means and hierarchical clustering, often face significant challenges when applied to continuously growing datasets. These limitations arise from factors such as restricted main-memory capacity, high computational requirements, and the temporal complexity of the underlying algorithms. As a result, traditional methods may become inefficient or impractical when processing massive datasets. This has created a strong need for scalable clustering techniques capable of handling large volumes of data efficiently while maintaining acceptable clustering quality.

Several strategies have been proposed to improve the scalability of clustering algorithms. Some approaches are inspired by techniques successfully employed in supervised machine learning, whereas others are specifically designed for unsupervised learning environments. One common strategy involves sampling, in which only a representative portion of the original dataset is selected for processing, thereby reducing memory and computational requirements. Another approach partitions the complete dataset into multiple smaller and relatively independent batches. Each batch is processed separately, and the resulting partial clustering models are subsequently combined to produce a final clustering solution.

Other techniques are designed for data-stream environments, where data arrives continuously and must be processed either in real time or in successive batches. The specific implementation of these scalability strategies depends largely on the underlying clustering algorithm and the characteristics of the dataset. Consequently, a wide range of methods has emerged, each offering different advantages in terms of memory consumption, computational efficiency, scalability, clustering accuracy, and processing time. Therefore, systematically analyzing these approaches and organizing their characteristics is essential for developing effective clustering methodologies for large-scale datasets.



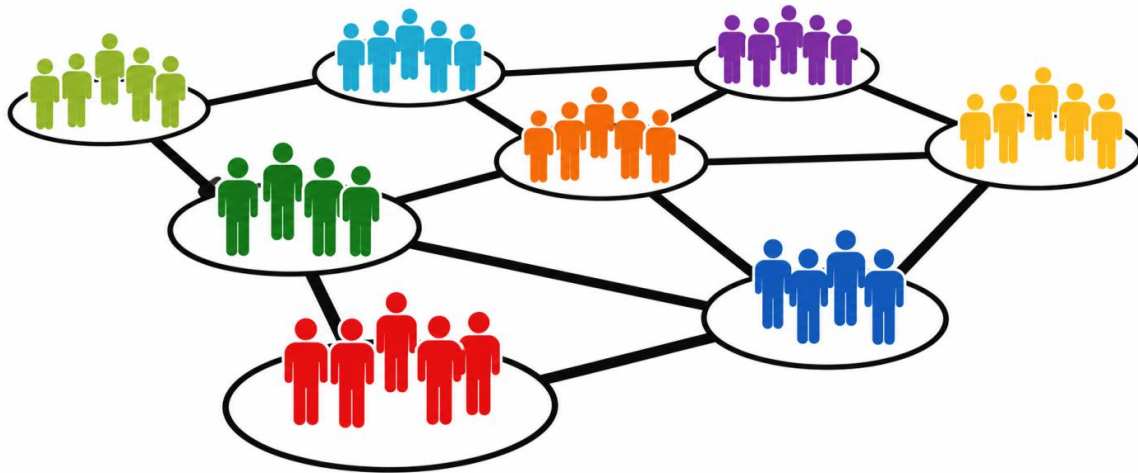


Fig. 1 Clustering

Cluster analysis is a fundamental technique in exploratory data analysis and has been widely applied in diverse areas such as image processing, speech recognition, information retrieval, and Web-based applications. The primary purpose of clustering is to organize a collection of data objects into meaningful groups, referred to as clusters, such that objects exhibiting similar characteristics are assigned to the same cluster, while objects with significant differences are placed into different clusters. From an optimization perspective, an effective clustering process aims to maximize the similarity or homogeneity among objects within the same cluster and maximize the dissimilarity or heterogeneity between objects belonging to different clusters.

Unlike classification, clustering does not depend on predefined class labels or a training dataset containing known categories. Instead, it discovers the inherent structure of the data based on the relationships and similarities among individual objects. Therefore, clustering is generally considered an unsupervised learning task in machine learning. The absence of predefined labels makes clustering a challenging problem. Different clustering algorithms may produce substantially different groupings for the same dataset. Furthermore, even when the same algorithm is employed, variations in parameter settings, initialization conditions, or the order in which data objects are processed can influence the resulting clusters. Consequently, selecting an appropriate clustering methodology and configuring its parameters are important factors in obtaining meaningful and reliable results.

Dataset Reduction and Dimensionality Reduction

Reducing the size and complexity of a dataset can be considered from two major perspectives. The first concerns the number of data instances or observations. When a representative subset of the dataset contains sufficient information to reflect the characteristics of the complete dataset, sampling techniques can be employed to reduce the amount of data that needs to be processed. Sampling can decrease memory requirements and computational cost while improving processing efficiency. However, this approach may not always be appropriate. In some applications, important information may be associated with relatively rare or infrequently occurring data instances. Such instances may be excluded during sampling, potentially causing the mining process to overlook valuable patterns or relationships.

The problem becomes more challenging when the data is generated continuously in the form of a data stream. In such environments, conventional sampling methods may be difficult to apply because the underlying data distribution can change over time. An inappropriate sampling strategy may therefore result in the loss of important or emerging information. Consequently, data-stream applications require sampling and processing mechanisms that can adapt to changes in the distribution and characteristics of incoming data.

The second important dimension of dataset reduction is the number of attributes or features. Large datasets often contain a high number of variables, many of which may be redundant, irrelevant, or strongly correlated. Dimensionality



reduction techniques are therefore employed to transform the original dataset into a representation containing a smaller number of informative features. The objective is to reduce the complexity of the dataset while preserving its essential structure and significant patterns. Ideally, analysis performed on the reduced representation should produce results that are comparable to those obtained from the original dataset.

In general, dimensionality reduction can be achieved through two primary approaches: feature selection and feature extraction. Feature selection identifies and retains the most relevant attributes from the original dataset while eliminating redundant or irrelevant features. In contrast, feature extraction transforms the original attributes into a new, smaller set of representative features that captures the most important information present in the original data. Both approaches can significantly reduce computational complexity and improve the efficiency of data mining and clustering algorithms when dealing with high-dimensional datasets.

II. REVIEW OF THE LITERATURE

Bozdemir et al. (2021) investigated a parallel clustering approach based on the Message Passing Interface (MPI) for processing educational datasets. The proposed M-K-Means approach combines MPI with the conventional sequential K-Means algorithm to distribute the clustering process across multiple computing nodes. In this approach, the original dataset is divided into p sub-datasets, where p represents the number of participating computational nodes, and clustering is performed independently on the distributed data. The approach was also evaluated using a DNA dataset with randomly selected initial centroids. The study demonstrated that conventional K-Means can provide satisfactory performance for relatively small datasets; however, its computational and memory requirements increase substantially as the volume and dimensionality of data grow. These limitations motivate the development of parallel and distributed clustering approaches capable of efficiently processing large-scale datasets. A related study by Shang et al. (2021) examined simple and parallel K-Means using education-sector datasets and discussed MPI-based M-K-Means, client-server parallelization, and other strategies for improving the processing of large datasets.

Wang et al. (2021) investigated a parallel K-Means strategy in which a threshold-based mechanism was employed to determine suitable initial centroids. Once the number of candidate elements reached a predefined threshold, the resulting values were selected as initial cluster centers. The study also discussed the ParaMeans software framework, which implements parallel K-Means using a client-server architecture. Such approaches aim to distribute computational workloads among multiple processing units and thereby reduce the execution time associated with conventional K-Means. However, the performance of parallel clustering remains dependent on factors such as data distribution, initialization of centroids, communication overhead, dimensionality, and workload balancing. Shang et al. (2021) further examined parallel K-Means and reported the use of client-server processing and dynamic load-balancing strategies for distributing subsets of data among computing nodes.

The problem of clustering very large databases has been investigated since the early development of scalable data-mining algorithms. Zhang, Ramakrishnan, and Livny (1996) proposed BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies) to address the limitations of conventional clustering methods when processing very large datasets. BIRCH uses a compact CF-tree data structure to summarize data and can incrementally process multidimensional observations while operating under memory and time constraints. The method was designed to reduce input/output requirements and can often obtain useful clustering results with a small number of scans of the dataset. This work established an important direction for scalable clustering by demonstrating that data summarization can substantially reduce the computational burden associated with large databases.

García et al. (2021) proposed a parallel batch K-Means approach for Big Data clustering. Their method partitions a large dataset into smaller batches and performs clustering on the individual partitions in parallel. The cluster centers obtained from the different partitions are subsequently combined and clustered to obtain the final solution. The approach reduces the amount of data that must be loaded into memory at a given time and improves the speed of clustering. Experiments conducted on multiple real datasets demonstrated the practical applicability of the method for large-scale data processing. However, the authors also highlighted challenges associated with determining an



appropriate batch size and ensuring that the clusters obtained from individual partitions adequately represent the complete dataset.

Li, Yang, and Han (2021) proposed a parallel K-Means algorithm based on the MapReduce framework for large-scale clustering. The approach divides the clustering task into partitioning, communication, combination, and mapping operations so that data can be processed across distributed computing resources. The study was motivated by the inability of traditional K-Means to efficiently handle large-scale marine datasets. Experimental analysis indicated that the MapReduce-based implementation could reduce computational requirements and improve processing efficiency compared with conventional K-Means.

Wang et al. (2014) investigated constrained clustering for Internet traffic classification. Their approach incorporated side information about relationships among traffic flows into a constrained Gaussian mixture model and used a constrained Expectation-Maximization algorithm. Experimental results using real-world packet traces showed that incorporating constraints could improve the quality of the resulting traffic clusters compared with conventional clustering approaches. This work demonstrates that additional information or constraints can be incorporated into unsupervised clustering to improve the usefulness of clusters in complex application environments.

Karwan Qader et al. (2017) compared three clustering-based data-mining techniques—K-Means, Fuzzy C-Means (FCM), and Expectation Maximization (EM)—for network fault classification. The study used network datasets representing both high- and low-traffic conditions and investigated the ability of clustering algorithms to identify abnormal network behavior. The results indicated that clustering could support automated network fault analysis and real-time fault-management applications. Although K-Means and EM reduced processing overhead, the computational requirements remained a concern for real-time environments. The study also highlighted the importance of suitable similarity measures for improving traffic clustering.

Qader and Adda subsequently investigated Fuzzy C-Means for network fault classification and reported that fuzzy clustering could provide useful classification of network faults while supporting visualization and decision-making. Their later work also explored optimized Fuzzy C-Means for detecting DoS and brute-force attacks, demonstrating the potential of clustering techniques for identifying abnormal behavior in network environments. These studies indicate that clustering can be effective in noisy and heterogeneous datasets, although computational efficiency and the selection of suitable clustering parameters remain important concerns.

Cohen-Addad, Mirrokni, and Zhong (2022) studied massively parallel K-Means clustering in the Massively Parallel Computation (MPC) model, which represents modern distributed computing environments such as MapReduce. Their work focused on developing scalable K-Means algorithms for perturbation-resilient datasets and demonstrated that, under appropriate data assumptions, highly parallel clustering can achieve strong approximation guarantees using a small number of computational rounds. This research further emphasizes the importance of distributed processing when clustering very large datasets.

More recently, **Sardar (2024)** reviewed the evolution of MapReduce-based clustering methods between 2013 and 2023. The review covered partitioning-based, hierarchical, and density-based clustering approaches and emphasized that conventional clustering algorithms often face computational and scalability limitations when applied to large real-world datasets. The study identified distributed clustering using MapReduce as an important research direction for addressing these limitations.

A 2024 survey on data-distribution strategies for parallel spatial clustering further emphasized the increasing importance of parallel clustering with the growth of Big Data. The study examined approaches based on distributed computing frameworks such as MPI, MapReduce, and Spark, highlighting that the way data is distributed among processing units can have a significant effect on clustering performance, communication cost, and scalability.

Guellil, Mohammed, and Keskes (2024) investigated distributed K-Means clustering using topological relationships. Their work addressed the limitations of centralized clustering methods in handling distributed data and explored the use of distributed processing to improve the scalability of K-Means. Such research demonstrates the continuing transition



from centralized clustering toward distributed approaches capable of handling geographically or computationally distributed datasets.

Research Gap

The reviewed studies demonstrate that K-Means remains one of the most widely used clustering algorithms because of its simplicity and relatively fast convergence, but its conventional implementation is not well suited to continuously growing and high-dimensional datasets. Existing research has addressed this problem through several strategies, including sampling, data partitioning, batch processing, MPI-based parallelization, MapReduce, Spark, data summarization, improved centroid initialization, and distributed computation. However, these approaches may introduce additional challenges related to communication overhead, load imbalance, initialization sensitivity, information loss during sampling, selection of batch size, and preservation of clustering accuracy.

Therefore, there remains a need for a scalable and efficient clustering approach for large datasets that can reduce computational and memory requirements while maintaining or improving clustering quality. In particular, an approach that combines dataset reduction with effective centroid selection and efficient clustering can potentially reduce the amount of data processed during each iteration without significantly compromising the quality of the final clusters. This research direction provides a strong motivation for developing an improved clustering methodology for large-scale datasets.

III. PROPOSED WORK

This research paper implements and examines Collection-based K-Mean (CBKM) algorithms, which are capable of finding clusters more quickly than normal K-means clustering methods. When the KMeans technique is used, the number of iterations required to converge the set of centres increases exponentially. Restarting MapReduce tasks, re-inputting the original data set, and shuffling intermediate results will occur repeatedly when large-scale data is clustered in parallel using the KMeans algorithm via MapReduce, considerably increasing both I/O and network expenses. As a result, a new collecting procedure is developed in order to minimise the amount of the input data. Simultaneously, the grid division approach will be considered a supplement to help reduce the time required to do a range query. The collecting algorithm will be enhanced, and representative verification will be considered. As the amount of data increases, the algorithm's time cost will become tremendous. The collection technique seeks to obtain a significantly smaller subset of points that accurately reflects all of the points and reduces the data size. Once we have the sample set, we will run KMeans on the sample points with representative verification. The collection algorithm can help reduce the size of the data and accelerate the clustering algorithm.

CBKM Algorithm

INPUT : Array $\{A_1, A_2, \dots, A_n\}$

A = data points

N = number of required clusteres

OUTPUT : a set of clusters

STEPS :

- 1) Randomly select A data points from dataset B as initial centers.
- 2) Calculate the distance between each data point A_i ($1 \leq i \leq n$) and all N clusters C_j ($1 \leq j \leq N$) and assign data set object A_i to the nearest cluster
- 3) For each cluster j ($1 \leq j \leq N$), recalculate the cluster center by taking arithmetic mean of each cluster
- 4) Repeat until no change in the center of cluster



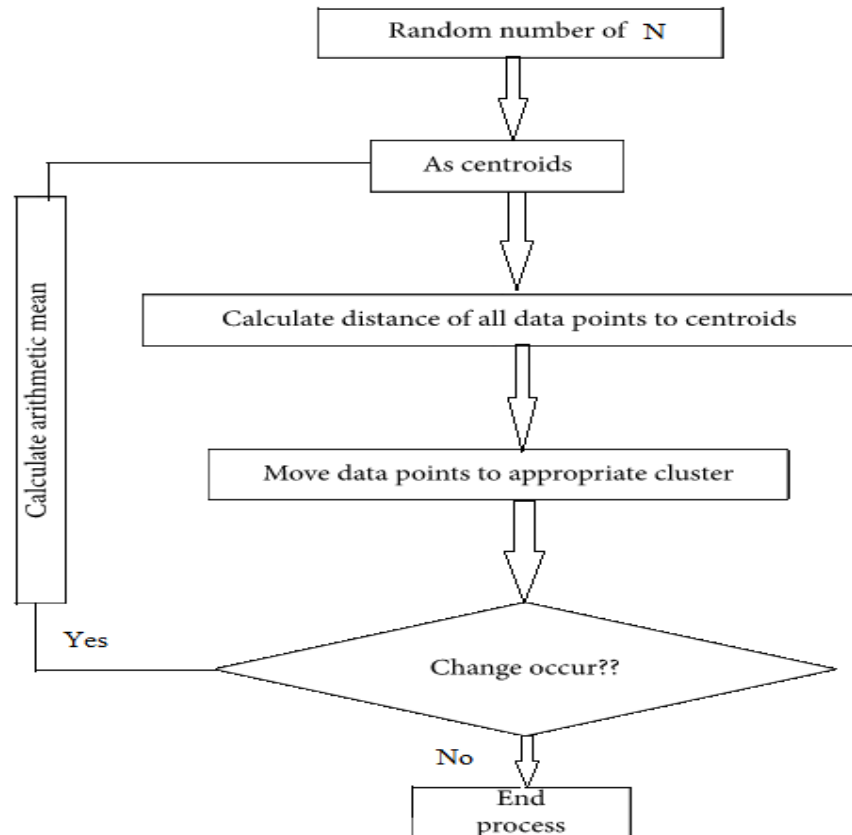


Fig. 2 Flowchart of CBKM

The clustering method begins with each input point as a separate cluster and combines the closest pair of clusters at each subsequent phase. To compute the distance between two clusters, c representative points are maintained for each cluster. These are found by selecting c evenly distributed points inside the cluster and then downsizing them by a fraction a' toward the cluster mean. The distance between two clusters is then calculated as the distance between the two clusters nearest pair of representative points - one from each cluster.

Thus, only the cluster's representative points are used to determine its distance from other clusters. The c representative points make an attempt to capture the cluster's physical structure and geometry. Additionally, via factor c_y , reducing the scattered points toward the mean eliminates surface anomalies and mitigates the consequences of outliers. This is because outliers are often located more away from the cluster centre, and as a result, shrinking causes outliers to migrate closer to the centre, while remaining representative points undergo modest alterations. The bigger movements of the outliers would therefore diminish their potential to integrate the incorrect clusters. Sampling is based on the assumption that just a random subset or subsets of the data are required to obtain a model for the data and that the entire dataset is available from the start. The random subsets may be disjoint or not. If multiple samples of data are processed, the subsequent samples are integrated into the present model. This is typically accomplished through the use of an algorithm capable of processing both raw data and cluster summaries. Because the algorithms that employ this method do not process all of the data, they scale according to the size of the sampling rather than the entire dataset.



IV. DISCUSSION & RESULTS

The performance of the proposed CBKM algorithm was evaluated by comparing it with conventional and existing clustering approaches, particularly standard K-Means and KM-HMR. For a given dataset, the input file is divided into multiple blocks according to the predefined block size, which is set to 64 MB by default. The proposed and comparative algorithms were then applied to datasets of different sizes to evaluate their effectiveness. A series of experiments was conducted using diverse datasets to assess important performance parameters, including clustering quality, execution time, computational efficiency, and scalability. The experimental evaluation was designed to determine whether CBKM could efficiently process large datasets while maintaining satisfactory clustering accuracy.

The datasets considered in the experimental study are presented in Table 1. One of the major datasets used in the investigation was obtained from Project Gutenberg (PG), a large digital library containing thousands of freely accessible electronic books, primarily consisting of public-domain works. The dataset provides a substantial collection of textual documents produced by numerous authors and therefore represents a suitable source for evaluating clustering techniques on large-scale unstructured data. Unlike structured datasets, unstructured text does not follow a predefined tabular format consisting of clearly defined rows, columns, and variables. Consequently, manually examining and categorizing a large collection of documents is extremely time-consuming and impractical.

Clustering provides an effective mechanism for organizing such documents according to their similarity. In this study, a subset of documents obtained from Project Gutenberg was selected for experimentation. To evaluate the scalability and effectiveness of the proposed approach, datasets containing different numbers of documents were prepared from the original collection, and the proposed algorithms were executed on each dataset. The selected documents represented a variety of subject areas, including audio, music, entertainment, children's literature, education, comics, crafts, finance, health, and markets. The diversity of these documents makes the dataset appropriate for evaluating the ability of clustering algorithms to identify meaningful groups within heterogeneous and unstructured textual information.

The experimental analysis of the Project Gutenberg dataset was particularly useful for examining the performance of the proposed approach in a large-scale text-mining environment. Since textual datasets can contain a large number of documents and features, conventional clustering methods may require substantial computational and memory resources. By applying CBKM to document collections of varying sizes, the study evaluates whether the proposed approach can reduce processing requirements while preserving the quality of the resulting clusters. The results obtained from these experiments provide a basis for comparing CBKM, standard K-Means, and KM-HMR in terms of efficiency, scalability, and clustering effectiveness.

Table 1 Datasets description

Dataset	Size (GB)	Description
DS_A: million song dataset	300	Collection of 53 audio features and metadata
DS_B: US climate reference network (USCRN)	200	Collected from 143 stations to maintain high quality climate observations
DS_C: Project Gutenberg	110	Includes over 50,000 free ebooks



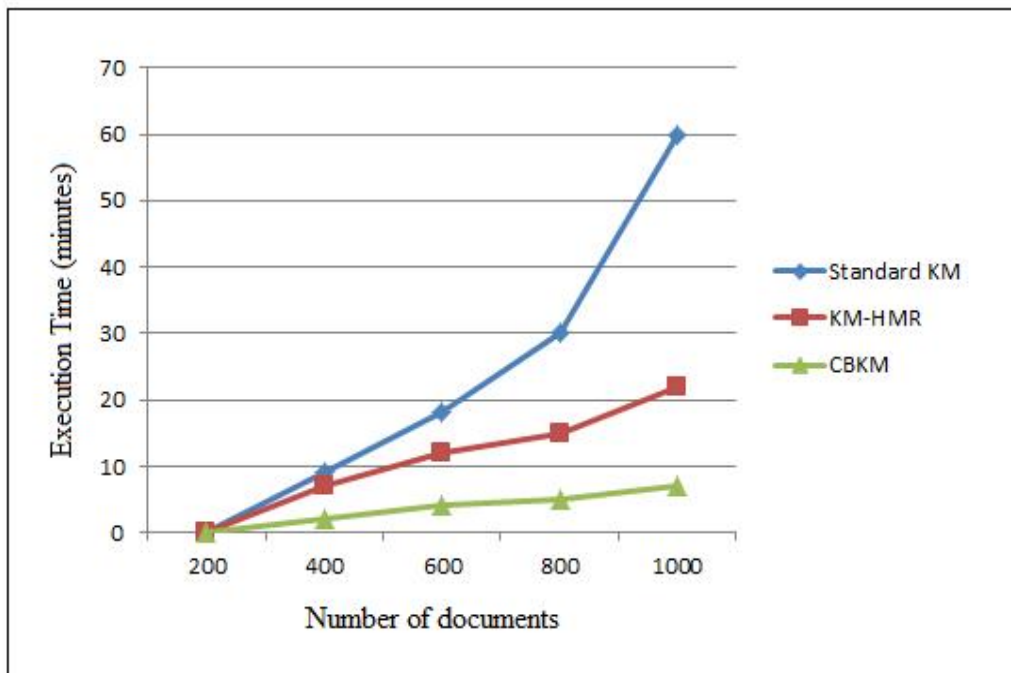


Fig. 3 Comparison of execution times with number of documents

The execution timings of the suggested algorithms for document sets from the PG dataset are shown in Figure 3. With an increase in the number of documents to cluster, the time required for standard K-means approaches the time required for the suggested solution. Parallel processing is critical when dealing with big amounts of data. We investigated only a part of the PG dataset in this experiment.

Table 2 Clustering results

Broad categories	Subcategories formed/total		
	KM-I2C	KM-HMR	Standard K- means
Digital audio	21/22	19/22	15/22
Space	26/28	17/28	14/28
Education	61/64	51/64	42/64
Entertainment	50/52	40/52	30/52

Table 2 displays Project Gutenberg's clustering results. The suggested CBKM algorithm requires less time to implement than the standard K-means algorithm because it processes large datasets across numerous machines. Whereas KM-HMR uses the Euclidean distance as a similarity metric, our suggested CBKM uses both intra- and inter-clustering measures to generate high-quality, effective clusters with low inter-cluster similarity and high intra-cluster similarity to extract meaningful information from big datasets.

V. CONCLUSION

The increasing volume and complexity of modern datasets have created a strong demand for scalable clustering algorithms capable of efficiently supporting unsupervised learning and data-mining applications. Conventional clustering and sampling techniques often encounter limitations when processing large datasets because of their computational time, memory requirements, and scalability constraints. These limitations highlight the need for



improved methodologies that can efficiently process growing data volumes without significantly compromising clustering quality.

To address the limitations of conventional sampling approaches, an improved aggregation-based method is proposed. The method constructs a representative sample set by aggregating data points from rectangular regions, sampling areas, and selected sample points. This strategy aims to reduce the volume of data requiring direct processing while preserving the important characteristics and distribution of the original dataset. Consequently, the proposed approach can provide a more efficient basis for clustering large datasets.

The proposed methodology provides a promising direction for improving the efficiency, scalability, and applicability of clustering algorithms in large-scale data-mining environments. Future work will focus on further improving the aggregation and sampling mechanisms, evaluating the approach on larger and more diverse real-world datasets, and investigating potential vulnerabilities and security issues associated with user-provided data. Additionally, future research can explore adaptive and intelligent sampling strategies that dynamically adjust to changes in data distribution while maintaining clustering accuracy.

REFERENCES

1. B. Bozdemir, S. Canard, O. Ermis, H. Möllering, M. Önen, and T. Schneider, "Privacy- preserving density-based clustering," 2021. View at: Google Scholar
2. L. Wang, H. Wang, W. Zhou, and X. Han, "A novel adaptive density-based spatial clustering of application with noise based on bird swarm optimization algorithm," Computer Communications, vol. 6, 2021. View at: Google Scholar
3. Karwan Qader, Mo Adda and Mouhammd Al-kasassbeh (2017), "Comparative Analysis of Clustering Algorithms in Network Traffic Faults Classification", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, No.4, pp. 6551-6563.
4. Yu Wang, Yang Xiang, Jun Zhang, Wanlei Zhou and BailinXie(2016), "Internet traffic clustering with side information", Journal of Computer and System Sciences, Elsevier, Vol. 80, No.5, pp.1021–1036.
5. M. J. Reddy and B. Kavitha, "Clustering the mixed numerical and categorical dataset using similarity weight and filter method," International Journal of Database Theory and Application, vol. 5, pp. 121–134, 2012.
6. Y. Wei, X. Zhang, Y. Shi et al., "A review of data-driven approaches for prediction and classification of building energy consumption," Renewable and Sustainable Energy Reviews, vol. 82, pp. 1027–1047, 2018.
7. M. Kumar, P. Chhabra, and N. K. Garg, "An efficient content based image retrieval system using BayesNet and K-NN," Multimedia Tools and Applications, vol. 77, no. 16, pp. 21557–21570, 2018.
8. A. Onan, S. Korukoğlu, and H. Bulut, "A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification," Information Processing & Management, vol. 53, no. 4, pp. 814–833, 2017.
9. Luiz Fernando Carvalhoa, Sylvio Barbona, Leonardo de Souza Mendes and Mario Lemes Proença (2016), "Unsupervised learning clustering and self-organized agents applied to help network management", Expert Systems with Applications, Elsevier, Vol. 54, pp.29-47.
10. P. Dahiya and D. K. Srivastava, "A comparative evolution of unsupervised techniques for effective network intrusion detection in hadoop," in Proceedings of the International Conference on Advances in Computing and Data Sciences, pp. 279–287, Dehradun, India, April 2018.
11. Mohiuddin Ahmed and Abdun Naser Mahmood (2015), "Novel Approach for Network Traffic Pattern Analysis using Clusteringbased Collective Anomaly Detection", Annals of Data Science, Springer, Vol. 2, No.1, pp.111–130.
12. S. Mehrotra and S. Kohli, "Comparative analysis of K-means with other clustering algorithms to improve search result," in Proceedings of the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), pp. 309–313, Delhi, India, October 2015.
13. Shital Salve and Sanchika Bajpai (2014), "Online stream mining approach for clustering network traffic", IJRET: International Journal of Research in Engineering and Technology, Vol. 3 No.2, pp.300- 304.



14. X. Zheng and N. Liu, "Color recognition of clothes based on K-means and mean shift," in Proceedings of the Intelligent Control, Automatic Detection and High-End Equipment (ICADE), pp. 49–53, Beijing, China, July 2012.
15. Lin Guan-zhou, XIN Yang, NIU Xin-xin and JIANG Hui-bai (2010), "Network traffic classification based on semi-supervised clustering", The Journal of China Universities of Posts and Telecommunications, Elsevier, Vol. 17, Supplement 2, pp.84-88.
- 16 Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, 103–114. DOI: 10.1145/233269.233324.
17. Shang, R., et al. (2021). Analysis of Simple K-Mean and Parallel K-Mean Clustering for Software Products and Organizational Performance Using Education Sector Dataset. Scientific Programming, 2021, Article 9988318. DOI: 10.1155/2021/9988318.
18. Li, Y., Yang, Z., & Han, K. (2021). K-Means Parallel Algorithm of Big Data Clustering Based on MapReduce PCAM Method. Engineering Intelligent Systems, 29(6), 411–418.
19. Cohen-Addad, V., Mirrokni, V., & Zhong, P. (2022). Massively Parallel K-Means Clustering for Perturbation Resilient Instances. Proceedings of the 39th International Conference on Machine Learning, 162, 4180–4201.
20. Sardar, T. H. (2024). Reflecting on a Decade of Evolution: MapReduce-Based Advances in Partitioning-Based, Hierarchical-Based, and Density-Based Clustering (2013–2023). WIREs Data Mining and Knowledge Discovery, 14(6), e1566.
21. Wang, Y., Xiang, Y., Zhang, J., Zhou, W., Wei, G., & Yang, L. T. (2014). Internet Traffic Classification Using Constrained Clustering. IEEE Transactions on Parallel and Distributed Systems, 25(11), 2932–2943.
22. Qader, K., Adda, M., Al-kasassbeh, M., et al. (2017). Comparative Analysis of Clustering Techniques in Network Traffic Faults Classification. International Journal of Innovative Research in Computer and Communication Engineering, 5(4), 6551–6563.

