Mesopotamian journal of Big Data
Vol. **(2023)** 2023, **pp**. 36–45
DOI: https://doi.org/10.58496/MJBD/2023/006 ISSN: 2958-6453
https://mesopotamian.press/journals/index.php/BigData

Research Article

Large Scale Data Using K-Means

Raheela zaib,*[1], OURLIS Ourabah[2],

[1]*Punjab university gujranwala campus, Pakistan*

[2]*University of Batna, Batna City, Algeria*

ARTICLE INFO

Article History
Received 22 Nov 2022
Accepted 03 Feb 2023
Published 13 Feb 2023

Keywords
Cluster analysis
Data mining
K-Means
KNN
Large Scale Data

ABSTRACT

Because of the exponential growth of high-layered datasets, conventional database querying strategies are inadequate for extracting useful information, and analysts must now devise novel techniques to meet these demands. Such massive articulation data results in a plethora of new computational triggers as a result of both the rise in data protests and the increase of elements/ascribes. Preprocessing the data with a reliable dimensionality reduction method improves the efficacy and precision of mining operations on densely layered data. Therefore, we have compiled the opinions of numerous academics. Cluster analysis is a data analysis tool that has recently acquired prominence in a number of different disciplines. K-means, a common parceling-based clustering algorithm, looks for a fixed number of clusters that can be identified using only their centroids. However, the outcomes depend heavily on the starting points of the clusters' focuses. Again, there is a dramatic rise in the number of distance calculations with increasing data complexity. This is due to the fact that assembling a detailed model typically calls for a substantial and distributed amount of preliminary data. There may be a substantial time commitment involved in preparing a broad collection of ingredients. For huge data sets in particular, there is a cost/benefit analysis to consider when deciding how to create orders: speed vs. accuracy. The k-means method is commonly used to compress and sum vector data, as well as cluster it. For precautious k-means (ASB K-means), we present No Concurrent Specific Clumped K-means, a fast and memory-effective GPU-based method. Our method can be adjusted to use much less GPU RAM than the size of the full dataset, which is a significant improvement over earlier GPU-based k-means methods. Datasets that are too large to fit in RAM may be clustered. The approach uses a clustered architecture and applies the triangle disparity in each k-means focus to remove a data point if its enrollment task or cluster it belongs to remains unaltered, allowing it to efficiently handle big datasets. This reduces the number of data guides that must be transferred between the CPU's Slam and the GPU's global memory.

1. INTRODUCTION

Data mining is a useful method for getting rid of designs since it addresses knowledge that is unquestionably hidden in huge data sets and shines a light on difficulties related to their potential, worth, feasibility, and adaptability. It may very well be considered a fundamental milestone during the knowledge disclosure period [1]. Data are typically preprocessed before being suitable for mining through data cleaning, data joining, data determination, and data alteration. Different data mining functions, such as class/idea portrayal, affiliation, connection analysis, organisation, prediction, cluster analysis, and others, provide suggestions for the kinds of examples that should be found. Many different databases and data stores can be used for data mining.

**Corresponding author. Email: Zzaib61@yahoo.com*

Cluster analysis is one of the crucial data analysis methods that is usually employed for the bulk of practical applications in developing nations. Clustering objects such that they are similar to (or linked to) one another and different from (or irrelevant to) the objects in other groups is the most common technique for finding groupings of objects. A successful clustering strategy will produce excellent clusters with high intra-cluster comparability and low between-cluster similarity. The way the approach is used, the similarity measure it employs, how successfully it detects some or all of the hidden samples and other factors affect the kind of a clustering outcome. A decent overview of clustering methods is available.

Choosing the primary k samples of the data focuses or randomly picking the underlying features are two straightforward techniques for handling cluster focus instatement. Another option is to choose the arrangement of beginning qualities that is closest to ideal among several beginning quality arrangements (from the data focuses). Additionally, there are much more distance estimates as the data complexity rises. Only a few factors are typically important to specific clusters as dimensionality rises, but information in immaterial components might create a lot of noise and hide the true clusters that are there to be found. Additionally, data typically grow more sparse as dimensionality rises, making data foci placed in various aspects appear to be equally far from one another and invalidating the distance measure, which is essentially worthless for cluster analysis. Therefore, characteristic reduce or dimensionality decrease is an important data-preprocessing activity for cluster analysis of datasets with a large number of items or credits.

An great amount of data is being generated by people as a result of the development of the web and the dissemination of new ideas, which makes mining and controlling it increasingly challenging. This massive growth of data also creates new challenges and requirements, which affects research across many different fields. Clustering techniques are used to really and successfully evaluate data structures [2]. However, how these data may be mined and grouped into intelligent groups is the most challenging question. By the way, several investigations have now been conducted to address this problem. On picture data, a focus on K-means, FKM, and IRP-K-means clustering techniques was led, and the presentation of these three algorithms was also investigated.

Another strategy combining K-Means and Hereditary Algorithm was developed to identify virtual entertainment data from local customers. Their framework's main goal was to cluster this social data by presenting the most efficient method for establishing the cluster centroid. Additionally, the clustering algorithm took into account three characteristics and used the advancement technique to produce exact clusters. Lashkari and Hussein put forth another approach that achieves the finest arrangements to address the problem of combination rate and global pursuit associated with K-Means. The analysis of the patient data was completed to group patients based on their EEG signal risk levels for epilepsy in order to evaluate the presentation of K-Means. During their evaluation, the obtained findings were also compared, and K-means and the KNN classifier both outperformed KNN. The semi-managed research that was conducted used a data set with a small number of noted occurrences. By applying the concept of parallelism in both computer chip and CUDA, the K-means algorithm update was completed. The examination that Praveen and Rama offered included more research on clustering tactics. Their assessment was delivered by outlining how effectively K-Means operates on the provided data set and delving into what the choice of the underlying seed gets meant for. K-means was also used to arrange geographical data with according to Hadoop. Another approach that considers K-Means and Fluffy C-means was proposed to work on the involvement of objects that were going to be demoted into clusters. This method allows data items to be distributed into clusters based on their degree of belongingness, which heavily depends on the selected fluffiness factor. By modifying the K-Means clustering technique using R-factual equipment, the analysis of multi-layered data related to the presentation of the understudy was completed.

2. LITERATURE REVIEW

Experts have made a few attempts to enhance the viability and efficacy of the K-means algorithm. Belal et al. (2005) [3] proposed an alternative strategy for cluster construction with the aim of identifying a group of medians away from a characteristic with the largest change. Although the approaches stated above can, in part, aid in discovering communities with extraordinary initialling, they are quite complicated and some of them make use of the K-means algorithm, which makes the employment of an unconventional methodology for cluster focus instatement necessary. Given the data parcelling method used for variety quantization, Deelers et al. (2007) [4] suggested a superior K-means method. Data is divided by the algorithm along the pivot with the most meaningful change.

2.1. K means clustering algorithm

The K-Means algorithm is one of the non-hierarchical partition-based clustering algorithms. The K-Means algorithm divides a collection of numeric elements X into sets of clusters k and attempts to minimize the sum of squared errors within the sets. In the k-means approach, we first construct k cluster habitats. The information data emphasis is then assigned to one of the existing clusters, and the closest cluster is selected based on the squared Euclidean distance between clusters. After each cluster mean (centroid) is processed, the cluster position is updated. This update is due to a change in

how each cluster is registered. The cycle of information vector diffusion and cluster position update is repeated until none of the cluster habitat values change.

The K-means algorithm's means are comprised of the following:

1. Establishment: Select K information vectors (data highlights) at random and then introduce the clusters.
2. Closest Neighbor Search: For each information vector, select the cluster community that is closest to it and give that input vector to the related cluster.
3. Mean update: Using the mean (centroid) of the information vectors allocated to each cluster, the habitats are updated.
4. Instructions for stopping: Keep repeating steps 2 and 3 until there is no longer any change in the mean value.

2.2. Analysis of Principal Components (PCA)

The Head Part Analysis technique was developed by Valarmathie et al. (2009) [5] and Yan et al. (2006) [6] to reduce the number of layers in a dataset while retaining the majority of the information that makes the dataset unique. As a mathematical transformation, principal component analysis (PCA) is defined as a symmetrical direct change that transforms the data into a different direction framework to the extent that any projection of the data results in the best variation on the primary direction (also called the main head part), the second most noticeable difference on the subsequent direction, etc. Therefore, PCA is a trustworthy method for uncovering crucial aspects in a multi-layered data set that explain differences in perspective. It can be used to study and depict complex data sets without sacrificing much detail.

2.3. Principal Constituent (PC)

The true definition of a vital component is a straight mix of ideally weighted detected components that increase the straight mix's difference and have no association with prior computers [7] . The largest overall change in the factors observed is indicated by a significant component of an important portion analysis. Two criteria of the additional parts that are not included in the study are that they are all overall uncorrelated with the beginning parts and that they each show the highest amount of observed variable volatility that was not covered by the earlier parts. The next parts will exhibit varied degrees of correlation with the observed components but will be entirely unconnected to one another after the main part analysis is complete.

Computers are recognised utilising the Eigen value decay of a data covariance network/relationship architecture or solitary value decay of a data lattice after generally mean-focusing the data for each quality. Covariance grids are chosen when factor differences are appreciably bigger than connectedness. It would be wiser to pick the form of cooperation when the factors are varied. In accordance with this, the SVD technique is employed to guarantee precision in mathematics.

2.4. Objectives

• To propose and select representative instances for multi-kernel learning and combine the advantages of the K-means clustering method and the outlier detection method to find the representative samples in a large-scale data set and the multi-kernel learning method to train an accuracy classifier.

• To determine the advantage of non-supervised method (i.e., K-means clustering, outlier detection) and supervised method (i.e., multi-kernel SVM) is combined, which produces satisfactory classification outcomes when using large-scale data set

3. METHODS AND MATERIALS

We provide a three-stage approach to achieve both a respectable accuracy and a rapid preparation speed. In the first step, the k-means clustering method is applied to a small subset of the initial full dataset. A larger percentage is more representative, but has a greater impact on clustering time. The proportion of cases selected has a large impact on both its representativeness and the overall speed of the algorithm. Our method chooses the percentage as conservatively as possible if the result is likely to be a sufficiently large variety of test classes for a given dataset. For the majority of our survey datasets, 10% of the baseline data is sufficient for clustering [8]. A K-Means clustering handle is constructed using a proprietary method that specifies a random clustering number k and runs K-Means periodically to recognize delegate occurrences as the underlying preparation set. In our methodology, RT represents the reputation time limit. The underlying data is then filtered based on the test exception score calculated using the anomaly location method. Subsequent mock

exams on naming will be made available to experts. The selected readiness set is slightly smaller after exception isolation, but the selected tests accurately represent the readiness data. Finally, the selected collection is used to develop a multi-kernel SVM classifier. Evaluations are made using the last remaining tests from the initial extensive data collection.

3.1. Selection of training cases

The results of K-means are strongly influenced by the predefined number of clusters k and the initial phase of clustering chosen, so running multiple K-means segments on the same dataset can capture different clusters. . Using this K-Means property using this method, we can find a more logical set of conditions that considers the distribution across the dataset. Much less human work is required to mark these selected cases for preparation. We describe a random cluster number to get different centers and edge centers for each k-means iteration. These focus regions cover the entire dataset required to build a highly accurate model. The K-Means method is iterated 5-30 times in the early stages of the approach. Therefore, the number of clusters in k-means is defined as an integer between 5 and 30. The nearest and farthest samples of each cluster community are collected. In this case, we use the power of the K-Means algorithm to combine different clusters in each phase in order to identify delegate events. These features can be used to consistently group the closest and furthest deposits close to the focal point. The selected situation is considered an instance prepared by the agent.

3.2. The Step of Outlier Discovery and Reduction

Duplicate versions of data in the original collected data collection are removed only to avoid frequent duplication. Some special situations can even make developing a classifier more difficult. Using the method proposed by Kim for outlier detection, we check the outlier score of each sample (2013).

To determine outlier tests, we record the Kolmogorov-Smirnov measurement between a given instruction j and another point in the collected set.

$$\mathrm{KS}(p_j - p_i) = \sup_{\mathrm{x}} |F_{p_j}(x) - F_{pi}(x)| \quad (1)$$

In the collected set, Fp j is the appropriate path leading from guide j to another point. The KSE test measurement will be processed using the normal of the Kolmogorov-Smirnov test insights, which will create the exception score in light of equation:

$$\mathrm{KSE}(p_j) = \frac{1}{n-1} \sum_{i=1}^{n} \mathrm{KS}(p_j - p_i) \quad (2)$$

The compared example will be handled as an exception and deleted in a basic manner if the usual KS measurement, here referred to as KSE measurement, exceeds a limit for a tested occurrence [9]. Finding a fair threshold is difficult. In order to repeat this interaction several times, The occurrences with the highest exception score are simply deleted. In our method, the repeated anomaly location time is used to characterise the recurrent time (Decay). A human master will then designate the reduced preparation set R as prepared before giving it to us.

3.3. Step 3.3: Test and train multi-core SVM

The selected dataset is labeled and then used as the training set to generate the classifier in the third step. Multi-core SVMs have been proven to outperform traditional SVMs. It takes longer to prepare a multi-core SVM when large amounts of data are used [10]. After the clustering and data reduction phases of our approach, the size of the selected occurrences is significantly reduced, making it more suitable for multicore preparation. Since it works faster and more efficiently than other multi-core learning strategies, we adopt the Simple MKL strategy as presented in Rakotomamonjy et al. (2008) to build the model.

3.4. Describe the algorithm

Table 1 gives a description of our algorithm.

TABLE I. PROPOSED ALGORITHM

```
Input: large-scale unlabeled data
Output: predicted label

Step 1 Clustering stage
1.1 select a small percentage of instances;
1.2 set the parameter of target cluster number of K-means as k=5 to 30;
1.3 set the clustering repeating sessions as RT=5 to 30;
1.4
    for i=1:RT
        use the K-means clustering method to cluster data into k classes;
        select the nearest and farthest instances to each cluster center
            and add them to the initial training set;
    end

Step 2 Outlier detection and reduction stage
2.1 delete repetitive instances in the initial training set;
2.2 compute the outlier scores of all instances;
2.3
    for i=1:ROT
        delete the instances with the highest score;
    end
    get the reduced training set;

2.4 label the reduced set;

Step 3 Training stage
3.1 set multi-kernel SVM parameters;
3.2 train the multi-kernel SVM classifier based on labeled training set;

Step 4 Evaluation
4.1 use the rest of instances to predict and test;
4.2 evaluate the performance.
```

4. RESULT AND DISCUSSION

The KEEL data set vault and the UCI AI storehouse are both used as the sources of the data sets used in the experiments in this section. All tests were conducted using the MATLAB R2015b software, which was powered by an Intel(R) Center i5-3470 computer processor running at 3.20 GHz and 4.0 GB of RAM.

During the examinations, we primarily assess accuracy and execution speed. Accuracy was defined as the ratio of perfectly sorted cases to the total number of test occurrences. The execution time of each strategy is equal to the duration of one processor season of the computer.

4.1. Performance Evaluation of Data Sets of Various Sizes

4.1.1. Setup for experiment

Wisconsin vehicle review, diabetic retinopathy, breast disease and four small to large sized datasets are taken from this section (bosom w) Debrecen While the dataset (messidor) comes from AI repository of UCI, spam base dataset comes from KEEL data warehouse. We compare our approach with Lib SVM [11].

Unlike traditional SVM (i.e. Lib SVM), our approach requires only a minimal number of instances to get started in order to achieve a respectable level of accuracy. such as selecting and describing a distinct k value to set the number of clustering for different datasets; K-means will promote the RT time and the furthest and closest versions for each cluster location will be selected; Outlier tests are also dropped from the selected set by outlier scores, which are then used to build a multi core SVM classifier. For testing, the remaining versions of each dataset are used.

4.1.2 Performance comparison

Accuracy and runtime for small to large datasets are collated in Table 2.

TABLE II. ACCURACY BASED ON SMALL TO LARGE SCALE DATA SETS

Data set	Samples	Training Instance Size		Accuracy (%)		Running time (s)	
		Proposed	***LibSVM***	***Proposed***	***LibSVM***	***Proposed***	***LibSVM***
Breast-w	683*10	20	457	81.33	81.64	2.52	0.26
Messidor	1151*20	35	830	50.32	53.11	3.33	0.65
Car	1728*6	55	2171	78.24	80.10	1.76	0.15
Spam base	4601*57	25	2570	66.64	66.33	8.28	23.40

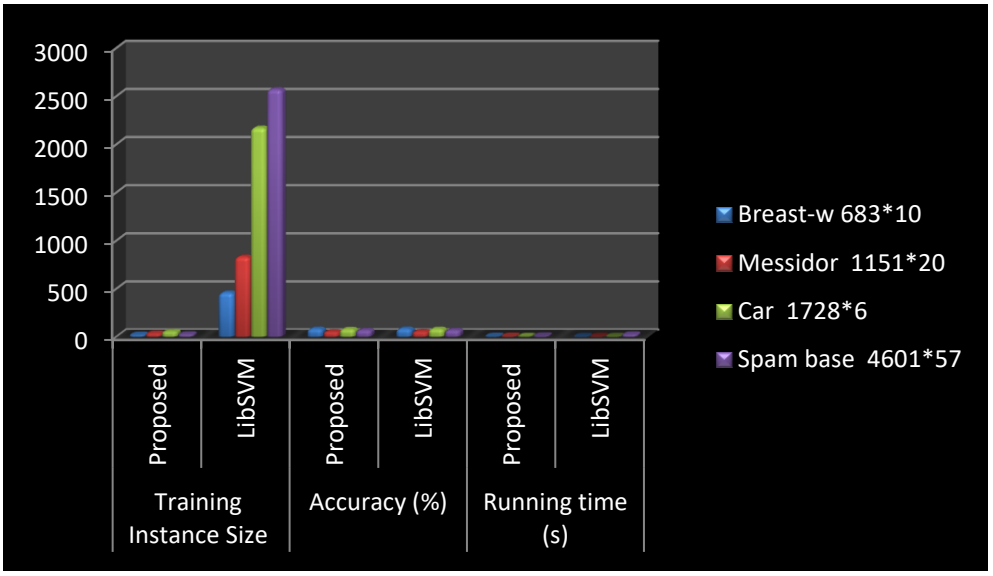

Fig. 1 accuracy based on small to large scale data sets

Numerous cases are required to prepare the traditional library SVM, which implies that marking the examples will take a lot of time and effort [12]. However, in our plan, we just need to designate a small portion of delegated preparation times without exact eroding. Therefore, the unrelenting labour of human manual marking is substantially diminished by our example choosing technique. Segments 5 and 6 of Table 2 show that, although requiring fewer delegate opportunities for preparation on various sizes of data sets than Lib SVM, our technique can nevertheless achieve a comparable level of accuracy.

The computational cost of K-means is O where T is the number of accents, k is the number of clusters and n is the number of items in the information data collection (Tkn). The computational cost to solve the SVM problem, where n is the number of occurrences, is both quadratic (n2) and cubic (n3) when C (the regularization boundary in equation 2) is minimal. Therefore, as can be seen in Section 7, the speed of our strategy is significantly slower than that of a conventional SVM when n is small. In any case, as the size of the dataset increases, our technique performs much faster than the default Lib SVM. The results show that even on small data sets, such as the data set w, messidor, and vehicles, our method can achieve accuracy and speed comparable to that of a Conventional SVM; however, as the size of the dataset increases, such as the spambase dataset with 4601 cases and 57 facets, the proposed method is both faster and more accurate than the Lib SVM. The accuracy and speed advantages of our strategy become apparent when organizing large datasets. The next review will see how well our method works with larger data sets.

4.2. Performance evaluation of extremely large data sets

4.2.1 Context for experiment

To demonstrate that our technique can be used for extremely large datasets, we examine five very large datasets. The coil2000 dataset pulls from the KEEL dataset archive, while the bank marketing database, skin split dataset, cover type dataset, and Containment data for individual action (keeps proteins) all come from the UCI datastore. First and foremost, we evaluate how quickly and accurately our method applies to very large data sets [13]. In addition, we test using several techniques.

4.2.2 Analysis of accuracy and speed

Table 3 shows how well our approach organised extremely large data sets. The percentage of cases utilised for clustering (proportion), number of clusters in the K-means clustering method (k), frequency of K-means clustering iterations (RT), and frequency of removing the exceptions with the highest anomaly scores are all included in Table 3 section 3 boundaries (Decay). After handling exception detection and clustering, few delegate occurrences are obtained. We list the frequency of delegate preparation events in section 4. It is just necessary to cite these examples. The benchmark is the real fact of the data sets.

TABLE III. PERFORMANCES FOR VERY LARGE DATA SETS

Data set	**Size**	**Parameters**	**Performance**						
			Training size	***Accuracy %***	***Execution time (s)***				
					Clustering	***Deleting***	***Training***	***Testing***	***Total***
Coil2000	9822*85	ratio = 0.02; k = 26; RT = 12; ROT = 26	57	83.62	0.28	2.50	3.60	0.23	4.02
Bank marketing	45,211*17	ratio = 0.1; k = 9; RT = 15; ROT = 5	24	77.33	1.44	0.03	0.16	0.06	3.98
Skin segmentation	245,057*4	ratio = 0.0005; k = 5; RT = 15; ROT = 5	27	82.32	0.06	0.04	0.26	0.52	2.20
Cover type (Aspen vs others)	581,012*54	ratio = 0.02; k = 10; RT = 30; ROT = 5	40	84.82	44.78	0.43	2.00	2.53	47.52

There aren't many examples of delegation planning that will really occur once the data reduction process was finished utilising the exception approach, as shown in Table 3 [14]. The first cover type data set has a range of occasions, but the preparation occurrence size is only 50, and the delegate test set is a very tiny subset of the first data, as shown in Table 3 section 4. The quantity of manual marking labour might be significantly decreased as a consequence. The selected agent case set advances the multi-kernel SVM approach while lengthening the preparation period.

4.2.3. Comparisons

For exceptionally large-scale datasets, Lin et al. (2015) proposed a framework called Delegate Data Recognition (ReDD), which is similar to our framework in that it performs case identification and classifier preparation in the next section. They compared their method with reference techniques that use GA, IB3 or DROP3 for example for identification and Truck, k-NN or SVM for characterization [15]. As can be shown in Table 4, when comparing their results to our own on the Confinement Data for Individual Movement (Protein Expectation) assessment data set, we discover that their speed and exactness are substantially slower than ours.

TABLE IV. COMPARISON WITH EQUIVALENT TECHNIQUES

Data size	**Average processing time 0**			**Average accuracy (%)**		
	Baseline	***ReDD***	***Proposed***	***Baseline***	***ReDD***	***Proposed***
145,751*74	2207.45h	252.28h	388.7s	85.24	89.46	88.32

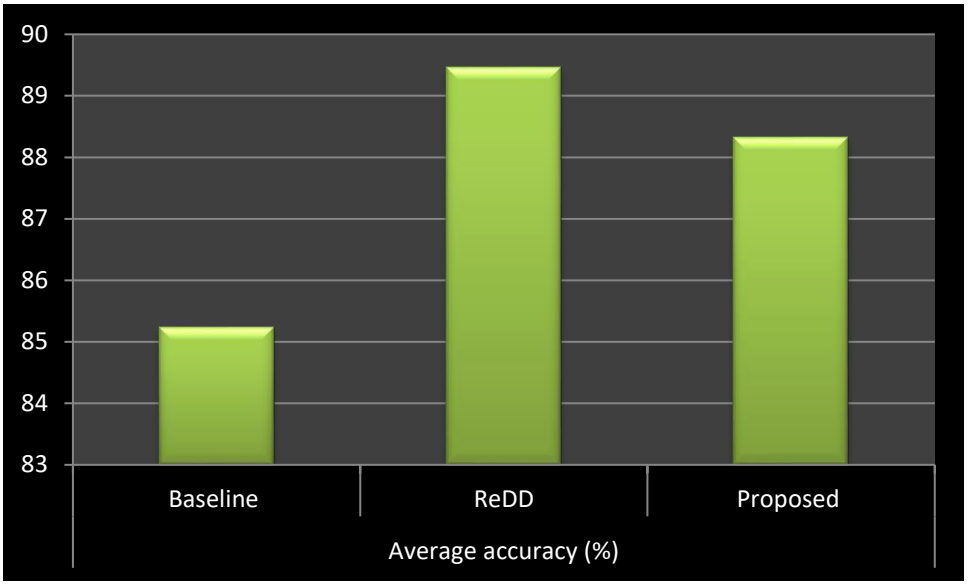

Fig. 2 comparison with equivalent techniques

Due to the amount of the data, either there is not enough RAM or it would take a long time to execute the LibSVM or ReDD strategies mentioned before. After clustering and exception localization in our technique, we will pick up a few delegate cases for multi-kernel preparation. In order to provide a connection and show the reliability of the recommended test determination technique, we compare our approach for developing the multi-kernel SVM with the traditional irregular choice methodology [16]. As shown by their ground truth, the remaining data sets are employed for testing and evaluating accuracy. We present the correlations in Table 5 together with the outcomes of the 15 assessment rounds for least exactness, maximum precision, and normal precision.

TABLE V. EXACTNESS TESTING USING AN ARBITRARY DECISION-MAKING APPROACH

Data set	**Data size**	**Training number**	**Random-selection method**			**Proposed method**		
			Min	***Max***	***Average***	***Min***	***Max***	***Average***
Coil2000	9822*85	57	77.00	82.80	82.24	98.76	83.02	81.62
Bank marketing	45,211*17	24	48.53	77.62	72.24	77.32	77.32	77.32
Skin segmentation	245,057*4	27	73.32	89.33	83.36	78.05	86.54	82.32
Cover type	581,012*54	40	2.52	89.26	72.04	84.73	85.62	84.82

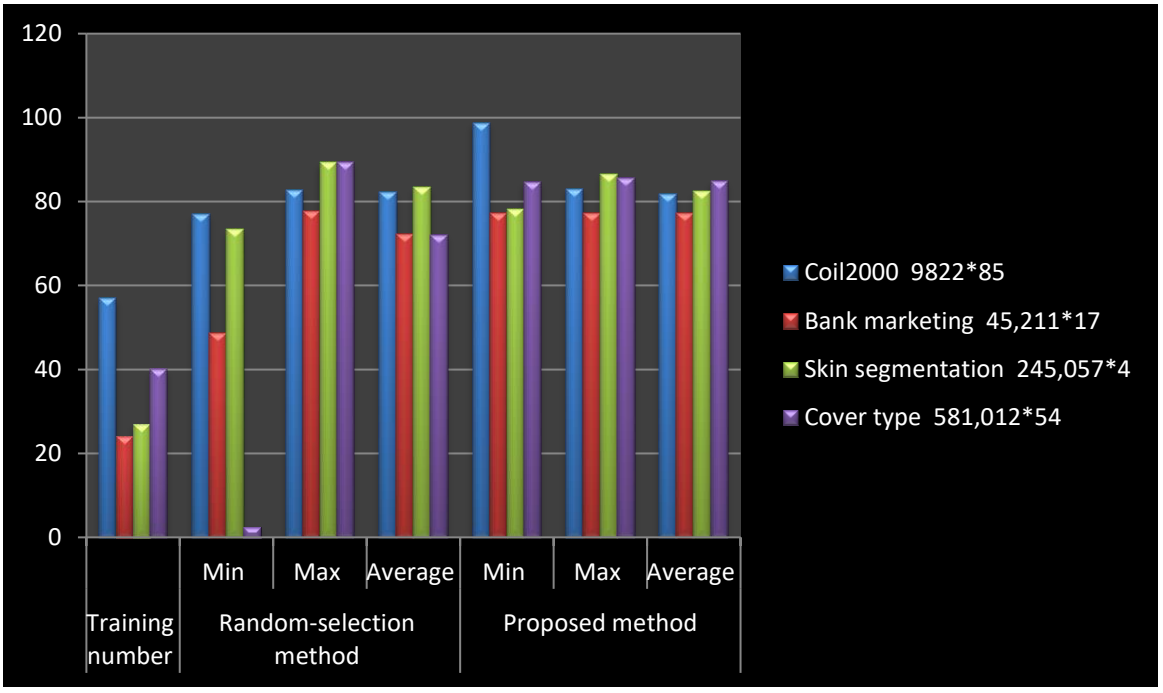

Fig. 3 Exactness testing using an arbitrary decision-making approach

Although not always reliable, the inconsistent preparation test determination approach can provide acceptable precision in some data sets [17]. Table 5 displays the vast variation in performance between the irregular strategy's minimum and maximum levels of precision. Our method, however, can produce significantly more trustworthy results. It is evident that our method can generate superior, more predictable precision.

5. CONCLUSION

The PCA method to reduce the size, a new approach to add cluster focus and the technique of assigning data focus to appropriate clusters are all combined in the proposed K-means algorithm. The proposed method is used to divide a certain data set into k clusters, with the aim of minimizing the total number of clustering errors for each cluster while keeping the embedding distance as high as possible. To meet the requirements for the simultaneous and accurate organization of very large-scale datasets, we have presented a three-step technique. By repeatedly running K-means with a customizable number of clusters, we use an uncommon strategy to select proxied samples in the first step. To refine the selected cases based on their outlier scores, an outlier search method is also applied. With this approach, the delegate preparation time allocated for manual naming will be significantly more modest. Finally, using the specified events, a multi-core SVM model is built. The demonstration of the proposed method is evaluated using data sets of different sizes. The outcomes demonstrate how our technique may be more accurate and efficient than traditional methods. Our technique achieves rapid and reasonable accuracy on characterization while considerably reducing the amount of human marking effort on exceedingly large data sets.

Conflicts of Interest

The author declares no conflicts of interest with regard to the subject matter or findings of the research.

Funding

The author's paper does not provide any information on grants, sponsorships, or funding applications related to the research

Acknowledgment

The author acknowledges the support and resources provided by the institution in facilitating the execution of this study.

References

[1] S. Bettoumi, C. Jlassi, and N. Arous, “Comparative Study of k-means Variants for mono-view clustering,” in International Conference for Signal and Image Processing -ATSIP, 2016, pp. 183–188.

[2] A. Alsayat, “Social Media Analysis using Optimized K-Means Clustering,” in IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA), 2016.

[3] Belal M. and Daoud A., 2005. A new algorithm for cluster initialization, World Academy of Science, Engineering and Technology, Vol. 4, pp. 74-76.

[4] Deelers S. and Auwatanamongkol S., 2007. Enhancing K-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance, International Journal of Computer Science, Vol. 2, No. 4, pp. 247- 252

[5] Valarmathie P., Srinath M. and Dinakaran K., 2009. An increased performance of clustering high dimensional data through dimensionality reduction technique, Journal of Theoretical and Applied Information Technology, Vol. 13, pp. 271-273

[6] Yan Jun, Zhang Benyu, Liu Ning, Yan Shuicheng, Cheng Qiansheng, Fan Weiguo, Yang Qiang, Xi Wensi, and Chen Zheng, 2006. Effective and efficient dimensionality reduction for large-scale and streaming data preprocessing, IEEE transactions on Knowledge and Data Engineering, Vol. 18, No. 3, pp. 320-333.

[7] Zhang Z., Zhang J. and Xue H., 2008. Improved K-means clustering algorithm, Proceedings of the Congress on Image and Signal Processing, Vol. 5, No. 5, pp. 162-172.

[8] M. Lashkari and M. Hossein Mottar, “The Improved K-means Clustering algorithm using the proposed Extended PSO algorithm,” in International Congress on Technology, Communication and Knowledge (ICTCK), 2015, no. Ictck, pp. 11–12.

[9] M. Manjusha and A. E. E. G. D. Acquisitiom, “Performance Analysis of KNN Classifier and K-Means Clustering for Robust Classification of Epilepsy from EEG Signals,” in International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016, pp. 2412–2416.

[10] X. Cui and F. Wang, “An Improved Method for K-Means Clustering,” in International Conference on Computational Intelligence and Communication Networks An, 2015, no. 1.

[11] M. Baydoun, M. Dawi, and H. Ghaziri, “Enhanced Parallel Implementation of the K-Means Clustering Algorithm,” in 2016 3rd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), pp. 7–11.

[12] P. Praveen and B. Rama, “An Empirical comparison of Clustering using Hierarchical methods and K-means Cluster Analysis Types :,” 2nd Int. Conf. on Advances Electr. Electron. Information, Commun. BioInformatics, pp. 1–5, 2016.

[13] Y. Zhong and D. Liu, “The Application of K-Means Clustering Algorithm Based on Hadoop,” in EEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), 2016, pp. 88–92.

[14] C. T. Baviskar and S. S. P. Associate, “Improvement of Data Object’s Membership by using Fuzzy K-Means Clustering Approach,” in International Conference on Computation of Power, Energy Information and Communication (ICCPEIC) Improvement, 2016.

[15] Agnivesh and R. Pandey, “Elective Recommendation Support through K-Means Clustering using R-Tool,” in 2015 International Conference on Computational Intelligence and Communication Networks Elective, 2015, pp. 851–856.

[16] Fahad A, Alshatri N, Tari Z et al (2014) A survey of clustering algorithms for big data: taxonomy and empirical analysis. IEEE Trans Emerg Top Computer 2(3):267–279

[17] Brodtkorb AR, Hagen TR, Sætra ML (2013) Graphics processing unit (GPU) programming strategies and trends in GPU computing. J Parallel Distrib Computer 73(1):4–13

[18] Hamerly G, Drake J (2015) Accelerating Lloyd’s algorithm for k-means clustering. In: Partitioned Clustering Algorithms. Springer, p 41–78

[19] He G, Vialle S, Baboulin M (2022) Parallel and accurate k-means algorithm on CPU-GPU architectures for spectral clustering. Concurr Computer: Pract Exp 34(14):e6621

[20] Kruliš M, Kratochvíl M (2020) Detailed analysis and optimization of CUDA k-means algorithm. In: 49th International Conference on Parallel Processing-ICPP, pp 1–11