STRATIFIED REMAINDER LINEAR SYSTEMATIC SAMPLING BASED CLUSTERING MODEL FOR LOAN RISK DETECTION IN BIG DATA MINING

[1]Kamlesh Kumar Pandey, [2]Diwakar Shukla

•

[1,2]Dept. of Computer Science & Applications,
Dr. Hari singh Gour Vishwavidyalaya, Sagar, M.P., India
[1]kamleshamk@gmail.com,[2]diwakarshukla@rediffmail.com

Abstract

Nowadays, large volumes of data generate by numerous business organizations due to digital communications, web applications, social media, internet of things, cloud and mobile computing. Such has turned the nature of classical data into big data. Loan risk analysis is one of the most importance financial tasks, where financial organizations predict loan risk through customer financial history and behavioral data. Financial institutions face loan risk related issues when they make a loan to a bad customer. As a result, financial institutions divide loan applications into loan risk and non-risk clusters before making a loan for avoiding the loan risk challenges. Clustering approach is a data mining technique that uses data behavior and nature to discover the unexpected loan without any external information. Clustering algorithms face efficiency and effectiveness challenges as a result of the primary characteristics of big data. Sampling is of the data reduction technique that reduces computation time and improves cluster quality, scalability and speed of clustering algorithm. This study suggests a Stratified Remainder linear Systematic Sampling Extension (SRSE) approach for loan risk analysis in big data clustering using a single machine execution. The SRSE sampling plan enhances the effectiveness and efficiency of the clustering algorithm by employing maximum variance stratum formulation, remainder linear systematic sampling and extending sampling results into final result through centroid distance metric. The performance of the SRSE-based clustering algorithm has been compared to existing K-means and K-means++ algorithms using Davies Bouldin score, Silhouette coefficient, SD Validity, Ray-Turi index and CPU time validation metric on risk datasets.

Keywords: Loan Risk Clustering, Big Data Clustering, Stratified Sampling, Remainder linear Systematic Sampling, Sample Extension, K-means, SRSE-K-means, SRSE- K-means++.

I. Introduction

The volume of data has increased rapidly as a result of the development of the internet of things, cloud computing, web applications, communication technologies and social networks. Big data mining is analysis and process dealing the massive amounts of data for an organization's decision-making system [1]. The major characteristics of big data are volume (large scale of data), variety (various categories of data), and velocity (speed of data, stay motion). These three Vs are referred to as core features of big data, whereas the remaining Vs are referred to as supportable characteristics. Veracity (quality of processed data), variability (inconsistency of data), value (importance of data) and visualization (imagining the data) are other characteristics. Volume is a key attribute of big data and is represented in the scale of Terabytes and Petabytes. Variety handles a wide range of heterogeneous data sources, formats and their types. Velocity represents the rate of data creation, generation, delivery and updates in batch time, real-time and streaming across the

heterogeneous sources [2–4]. Veracity determines the quality, trustworthiness and accuracy of the data during the mining process because some heterogeneous sources generate inconsistent, incomplete, imprecise and ambiguous data [3, 4] . Value related to attributes (importance) of data for decision-making during the analysis process. It is described as valuable information on a massive volume and heterogeneous data that does not impair business decisions [2, 5] . Variability indicates the nature of data across the time and is fragments used in big data sentiment analysis. It refers to data whose structure, meaning, and behavior constantly change over time due to rapid data growth [5, 6]. Visualization pictures the raw and analyzed data as per user expectation and understandable in the form of figure or graphical presentation such as a table, graph, picture, chart and so on [7].

Big data mining is the discovery of knowledge, unknown correlations, actionable information and hidden patterns in big data sources useful for decision-making [2]. The objective of data mining is to predict the unknown insight and provide a description of predicate values that users easily can interpret. A data relation approach is another way of big data mining that identifies the relationship between attributes of a dataset. Big data mining research necessitates transparency because the large volume of data provides valuable knowledge, relationships and hidden patterns. The variety of data types and data sources leads to a diversification of mining results, and data velocity defines real-time mining [8]. Big data mining utilizes stability, high-efficiency, low computational cost and better risk management capability [9]. The combination of statistics and data mining techniques is known as intelligent big data mining and addresses the process and management challenges in mining framework [2]. Big data mining under risk reduction is classified as association rule learning, clustering, classification, and regression prediction.

Clustering is a technique used risk reduction for unsupervised predictive data mining that predicts class label based on homogeneity, similarity, or characteristics. Each risk cluster has a high degree of resemblance and a significant separation degree among them. The distance between data points with the shortest distance within-cluster is defined as having a high similarity within-cluster. A high separation results in the maximum distance between clusters. [10]. The application of clustering is in the fields of pattern recognition, image segmentation, artificial intelligence, wireless sensor networks, text analysis, bioinformatics, financial analysis, vector quantization and so on [11, 12].

Clustering is used in risk analysis applications such as supplier risk assessment [13], probabilistic risk assessment [14], project interdependent risk [15], financial risk analysis [16], insurance risk analysis, dynamic rockfall risk analysis [17], fall risk assessment [18] etc. Credit and debit risk concentrations are managed by banks and financial departments. The clustering technique allows customers to spend less time processing loan applications, and financial organizations predict loan risk in terms of good and bad customer for loan repayment. Borrowers' loan repayment capacity and loan risk are determined by their liabilities, reliance on family members, loans from other sources, individual age, increase in future income, etc. The identification of loan risk factors improves organizational safety and performance [14].

Kara et al. [13] assessed the 17 qualitative and quantitative supplier risks using the K-means clustering algorithm. The data points within the cluster indicate the specific risk, and their interpretation facilitates risk management and reduces supplier risk. It used the supplier risk-related dataset to identify the most reliable supplier by minimizing risk. Mandelli et al. [14] used principal component analysis and mean-shift methodology to identify similar behavioral risk events using the clustering algorithm for probabilistic risk assessment. Marle et al. [15] used interaction-based clustering to categorize the risks. The proposed methodology used the clustering objective for prioritization and resource allocation during risk grouping. Kou et al. [16] used real-life credit and bankruptcy risk datasets to evaluate a clustering algorithm based on multiple criteria decision making (MCDM) problems for financial risk analysis.

Fahad et al. [19] outlined the volume, variety, and velocity evolution criteria of the

conventional risk clustering approach for big data. The volume of the conventional clustering technique is recognized as the dataset size, high dimensionality and outlier detection. The variety is recognized as a dataset type and the clustering shape of the conventional clustering algorithm. The velocity is considered in the complexity and execution time of the conventional clustering algorithm. The existing risk clustering algorithms are unsuitable for big data mining due to these characteristics in terms of scalability, performance, quality and speedup. Volume is a dominant attribute of big data that reason data mining algorithms to pose storage and processing challenges. Data Volume necessitates a large amount of hardware and takes a long time to execute algorithms. The most common big data clustering methods are incremental, divide and conquer, data summarization, sampling, efficient nearest neighbor, dimension reduction, parallel computing, condensation, granular computing and so on [11, 20–22].

Nowadays, sampling and distributed/parallelization systems are two major strategies to solve big data mining-related issues. Sampling is a widely scientific method in the context of big data because it accurately reduces the data amount to a manageable size, increases scalability and speeds up algorithm execution with data processing [23, 24]. The execution of risk clustering is divided into single and multiple machines categories under big data mining, where single machine clustering use single machine resources and multiple machines used distributed execution. Parallel/distributed computation and data reduction are two common approaches to large-scale data clustering [22].

Sampling is a data reduction strategy that is useful for improving efficiency and performance when dealing with various types of problems related to data mining and database systems [25–27]. Sampling process minimizes data size and saves computation time and memory, while establishing a balance between the computational cost of high volume data and approximation results [24, 28]. The sampling-based data mining technique reduces the amount of data for mining and is known as an approximation approach [22]. It achieves approximate results within a specific time with query optimization for the decision support system. It is used in high-volume data applications such as risk analysis, database sampling, online aggregation, correlation discovery, stream-sampling, and so on [29, 30].

The analysis of big data necessitates the use of highly scalable clustering techniques. The computational complexity of the classical clustering algorithms is high on large scale data set that reason it cannot be straight applied to large-scale. The computational efficiency and cluster quality are the major challenges in the large scale data clustering. The objective of this study is to improve computational efficiency in terms of scalability, resources utilization, computational cost, and speed-up of big data clustering utilizing stratified remainder linear systematic sampling extension (SRSE) approach in the application of loan risk analysis on single machine execution. This study is organized into five sections. The second section examines sampling-based clustering algorithms and their applications in data mining. The third section introduces the stratified remainder linear systematic sampling extension approach and provides a sampling strategy for big data clustering. Section four contains the proposed work implementation using the K-means and K-means++ algorithms and provides as well as their validation on loan risk datasets using internal measures. The final section of the work wraps up the work and explores new possibilities. The final section of the work concludes the work and explores additional possibilities.

II. Literature review

This section presents sampling-based works on data mining based on existing research perspectives and investigates the advantages of stratified and systematic sampling over other sampling methods. Most of the data mining algorithms use uniform random sampling, systematic sampling, progressive sampling, stratified sampling and reservoir sampling. Uniform random sampling selects data from large data sets using a random number generator [31]. In systematic sampling, the first data point of the sample is selected in random order and the remaining sample

data points are selected at fixed intervals from the dataset [32]. Progressive sampling starts with a small sample size and gradually increases the sample size until a satisfactory performance measure is obtained [25]. Stratified sampling splits the dataset into homogeneous sample data, which is known as strata, then uses random sampling to collect samples from the strata for processing [22]. Reservoir sampling is used for data stream mining for both homogeneous and heterogeneous data sources [33].

Buddhakulsomsiri et al. [34] used a stratified random sampling approach in the application of health care systems to bill processing accuracy. The sampling plan used the rectangular method for strata construction to utilize sample resources, and measured the accuracy by percent and dollar accuracy. Silva et al. [35] proposed the CLUSMASTER (CLUStering on MASTER) algorithm through sampling for data streams in the application of sensor networks. The sampling procedure shortens the execution time and allocates fewer resources to the MASTER algorithm. The CLUSMASTER selects the best samples from each sensor in a network while minimizing the sum of square errors of the cluster. Rajasekaran et al. [36] proposed the DSC (Deterministic Sampling-based Clustering) algorithm for hierarchical and partitional clustering. The DSC algorithm improved the speed and accuracy as compared to the random sampling.

Jaiswal et al. [37] proposed a PTAS method based on D2-Sampling and K-means clustering. The PTAS shortened the time required for exhaustive search and optimized the objective function of clustering. Parker et al. [36] introduced geometric progressive fuzzy c-means (GOFCM) and minimum sample estimate random fuzzy c-means (MSERFCM) accelerated algorithms. Both clustering methods used novel stopping criteria and sampling for subsample size identification to speed up the initialization process. The GOFCM algorithm combines single-pass fuzzy c-means (SPFCM) and progressive sampling, whereas the MSERFCM algorithm combines random sampling and fuzzy c-means extension.

Xu et al. [38] proposed the Summation-bAsed Incremental Learning (SAIL) algorithm to avoid effectiveness and efficiency issues associated with text clustering on a large scale of text documents. The SAIL algorithm employs random sampling to address data scalability issues using an approximate approach. The use of random sampling significantly reduces computation costs and controls sampling error. Luchi et al. [39] use K-means to cluster a large data set using random sampling and a genetic approach. This approach guides better sample selection through genetic operations and reasonable computing time.

Jing et al. [40] combined a stratified sampling method and an ensemble clustering algorithm on a high dimensional dataset. The stratified sampling is used to generate the subspace component of the dataset. The proposed method achieves a better clustering structure and more accurate results than random sampling and random projection methods without sacrificing cluster diversity. Li et al. [41] proposed a Distributed Stratified Sampling approach for big data. The stratified sampling extracts the subsample size from each partition of the data distribution in parallel order. The DSS algorithm achieved higher sample representativeness, accuracy, scalability, and efficiency with low data-transmission costs than state-of-the-art methods.

Zhan et al. [42] solved eigenfunction problems for spectral clustering algorithms in image segmentation applications using the Nyström sampling method. The Nyström technique is used to reduce the time and space complexity. The proposed method is effective for solving high-resolution image-related problems such as high dimensionality, small sample sizes, feasibility, and overall clustering solution. Aloise et al. [43] used an iterative sampling algorithm to solve the strongly NP-hard minimax diameter clustering problem (MMDCP). The proposed algorithm used the heuristic procedure to select the optimal solution across the sample.

Reddy et al. [44] proposed an optimal stratification design for data mining algorithms using Weibull-distributed auxiliary information in the context of a health population. The auxiliary information is used for strata construction in the absence of study variables. This study states that the combination of data mining and a well-designed sampling plan enhances the accuracy of mining results. Sainil et al. [45] compared the performance of stratified random sampling and

stratified ranked set sampling in terms of bias and mean square error. These evaluations show that stratified ranked set sampling is more efficient than stratified random sampling.

Li et al. [46] dveloped the clustering ensemble algorithm through sample stability, which divided the dataset into cluster core and cluster halo for the underlying cluster structure of the data set. The cluster core discovers the cluster structure through samples, and the cluster halo assigns the sample data into cluster construction. Zhao et al. [22] proposed the Stratified Sampling plus Extension FCM (abbr. SSEFCM) algorithm for large-scale datasets by combining stratified sampling and fuzzy c-means clustering. The SSEFCM improves computational efficiency and cluster quality while diminishing computational complexity.

Goshu et al. [47] proposed the Systematic Sampling Evolutionary (SSE) method, which is a derivative-free meta-heuristic type algorithm that combines a systematic sampling procedure and nature-inspired particle swarm optimization algorithm. Systematic sampling is used to determine the leader decision of the evolutionary algorithm, which searches for the action decision at each iteration. Prasad et al. [48] address the solution of the bigVAT algorithm through sampling and crisp partitions. The bigVAT is used for cluster tendency detection of big data clusters using the K-means algorithm on synthetic and real-life datasets. The sampling process selects a sample from inter-cluster data objects, and the crisp partitions technique predicts the cluster labels of sample objects.

Nguyen et al. [49] proposed the S-VOILA (Streaming Variance OptImaL Allocation) algorithm for streaming and non-streaming data using stratified random sampling and mini-batch processing. The S-VOILA algorithm reduces the variance of sample data through locally variance-optimal allocation and maintains the stratum via weighted sampling.

Larson et al. [50] investigated systematic and random sample designs and discovered that systematic sampling outperforms random sampling in terms of variance estimator, sample size, and data range. Stratified sampling outperforms simple random sampling in terms of statistical precision and sampling error. To achieve better accuracy, performance, and computing resource utilization, stratified sampling used a smaller sample size than random sampling [24]. According to the literature [41], stratified sampling can achieve higher statistical precision and improve representativeness by reducing sampling error than simple random sampling, because variability within subgroups with similar properties is lower than that of the entire population. Stratified sampling also extracts better samples from the dataset in terms of size and representativeness, which saves time and costs associated with the data processing algorithm.

The literature [32] states that sampled data from systematic sampling is more accurate and has spatial autocorrelation than random sampling. The results of the experiments [32] show that systematic sampling has variance-related issues that can be resolved by combining systematic and stratified sampling because each stratum has an optimal variance sample. The results of a comparison of uniform random sampling, progressive sampling, biased sampling, and stratified sampling show that stratified sampling achieves higher computational efficiency and quality for the clustering process [22].

III. Proposed Work

The practical approach of the sample plan for clustering across several domains is determined by existing research [22, 51, 52] and literature. The stratification technique reduces sample variance, whereas clustering reduces variance within a cluster. As a result, combining stratification and clustering improves the effectiveness and efficiency of clustering algorithm. Uniform random sampling is entirely dependent on sampling design, data structure, and sampling strategy. The random sampling does not cover the entire dataset; therefore the sample representativeness quality is reduced. To avoid this issue, systematic sampling is preferable because it sample data covers the entire dataset. This section describes the clustering objective, sampling contents, and presents the stratified remainder linear systematic sampling extension approach (SRSE) for loan

risk clustering on big data mining using single machine execution. The proposed method reduces computation costs and improves computational efficiency while maintaining cluster quality during risk clustering.

I. Objective function for loan risk clustering

Let the X loan risk based dataset $N = \{x_1, x_2, \ldots\ldots x_N\}$ to be clustered C into $K = \{C_1, C_2, \ldots\ldots C_K\}$ on the basis of predefined similarity function in d dimension space of loan risk attribute set. The considered clustering approach minimizes the within-cluster Sum of Squared Error (WSSE) and maximizes the between-cluster Sum of Squared Error (BSSE). The objective criterion defined described in Eq. 1 [20].

$$WSSE\ (X, C) = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \quad (1)$$

where x_i is the data point and μ_K is the centroid of C_K cluster. The content of C_K to the minimum SSE problem is defined by as under [53].

$$C_k = \left\{ x_i \in X | k = \underset{j \in \{1,2,\ldots.K\}}{\arg\min} \left\| x_i - \mu_j \right\|^2 \right\} \quad (2)$$

$$\mu_k = \frac{\sum_{x_i \in C_k} x_i}{|C_i|} \quad (3)$$

II. Sampling content

The presented clustering approach uses the stratification, remainder linear systematic sampling and sample extension process for loan risk group detection.

A. Stratification

The sampling-frame is divided into non-overlapping strata in stratified sampling according to data behaviors, types, location, attributes, variance, correction, regression, characteristics, format and so on. The strata are internally homogeneous with respect to the study variable that maximizes the precision of sampling results. Stratified sampling divides the N heterogeneous data points of loan risk dataset into $L = \{S_1, S_2, \ldots\ldots S_L\}$ homogeneous strata, where each stratum h consists of N_h data units and used the $\{S_1 \cup S_2 \cup \ldots\ldots \cup S_h\} = N$ and $\{S_1 \cap S_2 \cap \ldots\ldots \cap S_h\} = \theta$ conditions, where $h = 1,2,3, \ldots\ldots L$ and $\sum_{h=1}^{L} N_h = N = \{x_1, x_2, \ldots\ldots x_N\}$ [54, 55]. The stratum is derived from the loan risk data set by the stratification process. The maximum variance attribute and ascending sorting heuristics have used in this study to employ novel stratification methods. Algorithm 1 and Figure 1 illustrate the conceptual stratification representation.

This study used a remainder linear systematic sampling approach; therefore the stratification process formed the dataset into two strata. The stratification process first extracts the study variable based on maximum variance and then arranges the entire loan risk dataset based on the selected variable. The remainder linear systematic sampling method is used to determine the number of data points in strata.

B. Remainder Linear Systematic Sampling

Chang et al. proposed the Remainder Linear Systematic Sampling (RLSS) method to overcome the limitations of linear systematic sampling [56] in terms of *N≠nl* and linear sample size. Where N is

the size of the dataset, n is the sample size and l is the sample interval. The RLSS resolved the systematic sampling issue through a combination of stratification and linear systematic sampling. The number of data points in the loan risk dataset is represented in the RLSS approach by $N=nl+r$, $0\leq r\leq n$. The RLSS approach is more efficient when the sample size is not a multiple of the dataset size and in $N\neq nl$ situations. The n, l and r are the integer numbers, and r is the reminder data points of the sampling process [56][57] .

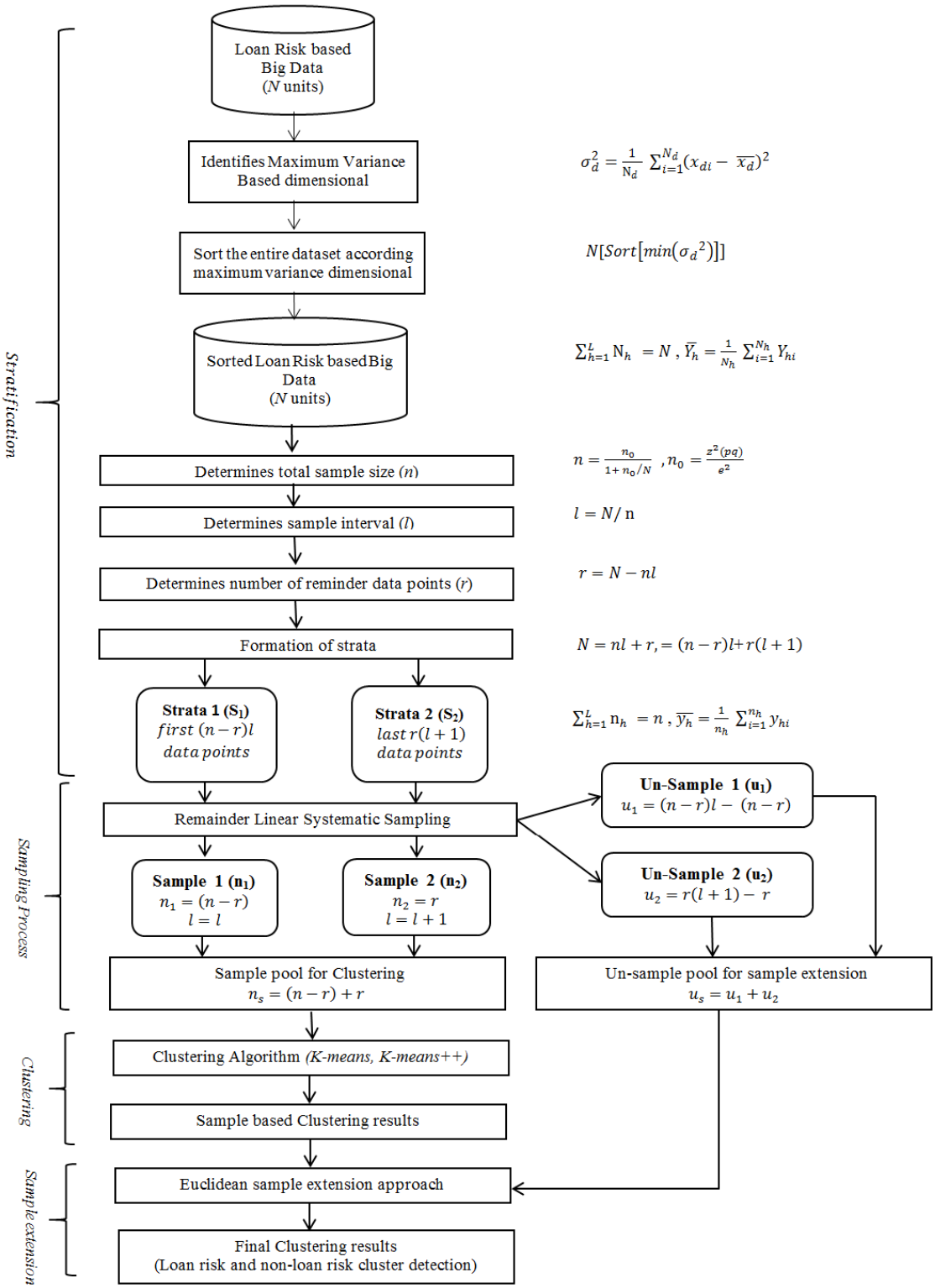

Figure 1. Conceptual Representation of proposed Stratification and Sampling Plan (SRSE)

The number of data points in strata is determined by the integer constraints *n, l,* and *r*. This study has adopted the Cochran formula for sample size identification, which is shown in Eq. 4 and Eq. 5. [58].

$$n_0 = \frac{z^2(pq)}{e^2} \tag{4}$$

where z denotes the standard error, p indicates the variability of the dataset, q signifies the *(p-1)*, and e represents the acceptable sample error. In this study, the standard error is set at 99% for the confidence interval, so the z value is set at 2.576, the variability value p was set at 0.5, and the acceptable sample error e is set at 1% for the 99 confidence interval. To obtain the total sample size, the sample size is normalized by the total number of data points in the loan risk dataset [58].

$$n = \frac{n_0}{1+ n_0/N} \tag{5}$$

The sampling interval is determines every n^{th} data point of stratum is chosen for clustering. Eq. 6 is determined the sampling interval.

$$l = N/n \tag{6}$$

The reminder data points *r* refer to the un-sample data points after the sampling procedure. Eq. 7 describes the identification of the number of reminder data points *r*.

$$r = N - nl \tag{7}$$

The number of data points in the strata is determined by the values of *n, l,* and *r*. The first stratum is made up of the first *(n-r)l* data points from the sorted loan risk dataset. The second strata is represented by the remaining *r(l+1)* data points of the sorted loan risk dataset. These scenarios are described in Eq. 8.

$$N = nl + r = (n - r)l + r(l + 1) \tag{8}$$

After stratification, the RLSS is used to define the sample size and sample selection interval for each stratum. The *(n - r)* data points of the first strata are selected for a sample pool/clustering with a *l* linear systematic sampling interval, while r data points of the second strata are selected for a sample pool/clustering with a *(l +1)* linear systematic sampling interval. As a result, Eq. 9 determines the total number of sample sizes n.

$$n = (n - r) + r \tag{9}$$

C. Sample extension

The Sample extension approach uses centroid-based distance to convert sample-based clustering results into final clustering results. The centroid-based distance used the Euclidean distance approach, which assigns un-sample data to its closed cluster using a centroid of the sample based cluster. Eq. 10 describes the sample extension function, where A_i is the data point of the un-sample data pool and B_i is the mean of the cluster centroid [22].

$$dis_{\text{euclidean}}(A, B) = \sqrt{\sum_{i=1}^{n} |A_i - B_i|^2} \tag{10}$$

III. Algorithm Description

This section describes the stratified remainder linear systematic sampling extension (SRSE) approach for loan risk analysis through stratification, remainder linear systematic sampling, and sample extension. The standard sampling plan first selects the dimension with the highest variance of the risk-based dataset and then sorts the entire dataset based on the selected dimension. The data points are then assigned to both strata using remainder linear systematic sampling rules. After the stratification process, it collects the required number of sample data points into a sample pool for clustering. The strata sample size and sample interval determined the sample data. The sample based clustering results are merged into the final results with the help of the sample extension method. The sample unit is used for clustering, and the resulting value is merged with an un-sample unit via sample extension. The proposed SRSE sampling plan is detailed in Algorithm 1 and the sampling flowchart is shown in Figure 1.

Algorithm 1 Stratified Remainder linear Systematic sampling Extension (SRSE) Big Data Clustering Approach

Input:

1. $N = \{x_1, x_2, \ldots \ldots x_N\}$ *is the data points of the loan risk based D dataset.*
2. *K = Required number of clusters.*

Output:

1. $C_K = \{ C_1, C_2 \ldots \ldots C_k \}$ *of the clustering results.*

Methods:

Stratification

1. Identify the maximum variance dimension of the dataset.
- $v_d = max(\sigma_{1d}{}^2, \sigma_{2d}{}^2, \ldots \ldots \ldots \ldots \ldots, \sigma_{Nd}{}^2,)$

2. Sort the entire data of the dataset according to the v_d dimension in ascending order.

3. Determine the total sample size *n* for the clustering process through Eq. 5.

4. Extracts sample interval *l* from the entire dataset by Eq. 6.

5. Define the number of remainder sample data points *n* through Eq. 7.

6. Determine the number of data points for each stratum with the help of *n, l* and *r*.
- $S1 = (n - r)l$
- $S2 = r(l + 1)$

7. Extract two strata from the entire dataset based on the sorted dataset .
- *S1=N[0:* $(n - r)$]
- *S1=N[*$(n - r): len(N)$]

Sample size identification

8. Determine the number of data point of sample size for both strata according to Eq. 9.
- $n_1 = (n - r)$
- $n_2 = r$

Sample allocation

9. Extract every l^{th} data points for *S1* and every $(l+1)^{th}$ data point for *S2* strata through linear systematic sampling.

10. Combine all *n1* and *n2* sample data points into the n_S sample pool and all un-sample data points into the u_S un-sample pool.

Clustering algorithm

11. Apply necessary clustering algorithms in n_S and achieved approximate clustering results such as *K-means* (n_S *, K), K-means++* (n_S *, K),* etc.

Sample extension

12. According to Eq. 10, assign u_S un-sampled pool data to approximate clustering results based on nearest Euclidean distance.

13. Achieved final clustering results in the loan risk and non-loan risk clusters and Exit.

IV. Experimental Analysis over Loan Risk Data

The experimental study evaluates the research effort based on the computing environment, datasets, existing algorithms, evaluation criteria, and outcomes. This section discusses the experimental environment, loan risk dataset characteristics, and validation criteria. The effectiveness and efficiency-related assessment criteria are used to evaluate the performance of the SRSE-based clustering approach.

I. Experiment Environments and Loan Risk Dataset

The computing environment of the SRSE-based clustering approach used in the Jupyter Notebook computing environment. The experimental environment is configured with an Intel I3 processor, CPU M350@2.27 GHz, 320 GB hard disk, 4(+64) GB DDR3 RAM, Windows 7 operating system, and Python tools. The experimental analysis was performed on four loan risk datasets within a single machine execution environment. Table 1 illustrates the characteristics and sources of the experimental loan risk datasets.

Table 1 Description of the Loan Risk Datasets.

ID	Datasets (DB)	Objects	Attributes	Class	Data Source
LRDB1	Bondora Peer to Peer Lending Loan Data	1,79,235	112	2	https://www.kaggle.com/
LRDB2	Vehicle Loan Default Prediction	3,45,546	41	2	https://www.kaggle.com/
LRDB3	XYZ_Corp Lending Data	8,55,969	70	2	https://www.kaggle.com/
LRDB4	Loan Data for Dummy Bank	8,87,379	30	2	https://www.kaggle.com/

The clustering of the LRDB1 loan risk-related dataset is divided into two classes: default risk and non-default risk. Default risk is a significant risk factor used to evaluate borrowers' behavior in peer-to-peer (P2P) lending. Lenders want to minimize the default risk on each lending decision in order to make rational decisions and to realize a return that compensates for the risk.

The loan risk-related dataset LRDB2 is clustered in order to estimate the determinants of vehicle loan default risk and non-default risk. The clustering process predicts the likelihood of a loanee/borrower defaulting on a vehicle loan during the first EMI (Equated Monthly Instalments) due date. This ensures that clients who are capable of repayment are not turned down. The important determinants are identified, which are used to reduce default rates.

The LRDB3 loan risk dataset clustering manages credit risk by using historical data to determine who to lend to in the future based on default, payment information, credit history, and other factors. The clustering process categorizes the data as capable of loan repayment or incapable of determining loan eligibility.

The LRDB4 clustering divides the data into loan default risk and non-loan risk. Data grouping provides funds for potential borrowers, and banks earn a profit based on the risk they take (the borrower’s credit score).

II. Selected Algorithms for Comparison

The proposed SRSE-based clustering approach is compared to partitional K-means (KM) [20], K-means++ (KM++) [59, 60] clustering algorithms. The goal of both clustering algorithms is to recognize loan risk in terms of default risk by minimizing within-cluster Sum of Squared Error (WSSE) and maximizing between-cluster (BSSE) distance. Except for the initial centroid selection approach, the cluster formulation process of both methods is similar. The KM method chooses the initial centroid at random, whereas the KM++ method chooses the initial centroid based on distance and probability.

III. Evaluation Criteria

Cluster validation is achieved through the application of both internal and external measures. The internal measure is used to compare the cluster's objective to its internal structures. The external measure is used to validate the cluster using outside knowledge. This study employs the Davies Bouldin score (DB), Silhouette coefficient (SC), SD Validity (SD), and Ray-Turi index (RT) as internal validation tools for effectiveness [61–63], with CPU time (CT) serving as an efficiency validation metric [64–66]. The strongest clustering method always maximizes intra-class similarity while decreasing inter-class similarity. As a result, the clustering method maximizes the SC metric value while reducing the DB, SD, RT, and CT metric values.

- Davies Bouldin score (DB): The Davies Bouldin validates within-cluster dispersion and between cluster similarity independently number of cluster. In the DB formulation, $|C_j|$ defines the total number of data point x_i inside of C_j cluster and C_i is another cluster.

$$DB = \frac{1}{K}\sum_{i=1}^{K} max_{i\neq j} \frac{within_i + within_j}{between_{ij}} \tag{11}$$

$$within_j = \frac{1}{|C_j|}\sum_{i=1}^{|c_j|} || x_i - C_j ||^2 \tag{12}$$

$$between_{ij} = || C_i - C_j ||^2 \tag{13}$$

- Silhouette coefficient (SC): The Silhouette coefficient validates cluster similarity by accepting the pairwise difference of cluster distances within (compactness) and between (separation) the clusters. In SC formulation *a(x)* is the average distance of *x* to all other data points in the same cluster *C*, *b(x)* is the average distance of *x* to all other data points in the all *Ci* cluster.

$$S = \left\{\sum_{x\in Ci} \frac{b(x)-a(x)}{\max[b(x),\ a(x)]}\right\} \tag{14}$$

- SD Validity (SD): The SD Validity metric assesses the effectiveness of clustering by averaging dispersion and total separation between clusters with variance. In SD Validity formulation, $\propto$ is constant value equal to 1, S_a is average scattering in term of variance and S_t is the total separation of cluster, $\sigma(C_i)$ is defines variance of C_i cluster, $\sigma(X)$ is represents variance of dataset.

$$\text{SD} = \propto S_a - S_t \tag{15}$$

$$S_a = \frac{1}{k}\sum_{i=1}^{k}\frac{\|\sigma(C_i)\|}{\|\sigma(X)\|} \quad (16)$$

$$S_t = \frac{D_{max}}{D_{min}}\sum_{i=1}^{k}\left(\sum_{j=1}^{k}\| C_i - C_j \|\right)^{-1} \quad (17)$$

$$D_{max} = max_{1\le i,j\le k} \| C_i - C_j \| \quad (18)$$

$$D_{min} = min_{1\le i,j\le k} \| C_i - C_j \| \quad (19)$$

- Ray-Turi index (RT) : The Ray-Turi index measures the mean of the squared distances of the all data points respect to k cluster centroid and minimum squared distance $\Delta_{kk'}^2$ between all cluster centroid. In the RT formulation, N is total length of dataset, M_i^k is the data points of particular cluster k and G^k is the centroid of that cluster. $G^{k'}$is the centroid of remainder cluster.

$$\frac{1}{N}\sum_{k=1}^{K}\sum_{i\in I_k}\|M_i^k - G^k\|^2 = \frac{1}{N}\sum_{k=1}^{K} WGSS^k = \frac{1}{N} WGSS \quad (20)$$

$$\min_{k<k'} \Delta_{kk'}^2 = \min_{k<k'} d(G^k, G^{k'})^2 = \min_{k<k'}\|G^k - G^{k'}\|^2 \quad (21)$$

$$RT = \frac{1}{N}\frac{WGSS}{\min_{k<k'} \Delta_{kk'}^2} \quad (22)$$

- CPU time (CT): CPU time computes the total execution times of any algorithm inside the CPU between the entry ENT and exit EXT times of the clustering algorithm.

$$CT = EX_T - EN_T \quad (23)$$

IV. Experimental Results and Discussion

On the basis of effectiveness and efficiency indices, the performance of SRSE-based clustering algorithms such as SRSE-KM and SRSE-KM++ has been compared to that of the classical KM and KM++ algorithms. Tables 2-3 show the average comparative efficiency and effectiveness results from four loan risk data sets using ten trials. This study used pre-defined Python library functions for DB, SC and SD, as well as technical code for RT, WSSE, BSSE and CT for cluster evaluation. Tables 2-3 highlight the optimal value of each reported result in bold face, where the optimal value of SC is required for maximization and the optimal values of DB, SD, RT, and CT are required for minimization.

Table 2 shows that the proposed SRSE clustering strategy outperformed the KM and KM++ algorithms in terms of WSSE, compaction, separation, similarity, dissimilarity, variance, and density. Table 3 demonstrates that the proposed SRSE strategy is faster than the KM and KM++ algorithms and uses the least amount of CPU time.

The experimental results of the LRDB1, LRDB2, LRDB3, and LRDB4 loan risk datasets illustrate that SRSE-KM and SRSE-KM++ clustering strategies outperform KM and KM++ algorithms in DB, SC, SD, and RT. In terms of clustering quality, the observed DB, SC, SD, and RT values show that the SRSE-KM and SRSE-KM++ algorithms outperform the KM and KM++ algorithms. Inside the LRDB1 risk dataset, the SRSE-KM diminishes the CT to 73.82% as compared to KM, and the SRSE-KM++ decreases the CT to 84.71% than KM++. Over the LRDB2 risk dataset, the SRSE-KM reduces the CT by up to 79.33% compared to the KM, whereas the SRSE-KM++

minimizes the CT to 90.48% than the KM++. For the LRDB3 risk dataset, the SRSE-KM alleviates the CT by 88.06% then KM while the SRSE-KM++ depletes the CT up to 93.79% in the context of the KM++. In LRDB4 risk dataset efficiency observations, the SRSE-KM minimizes the CT to 78.63% with reference to KM, and the SRSE-KM++ reduces the CT to 95.44% than KM++.

Table 2– Comparative average analysis of effectiveness measures ($\boldsymbol{means \pm std}$) over 10 trials

DB	Criteria	KM	**SRSE-KM**	KM++	**SRSE-KM++**
LRDB1	DB	1.93012 ± 0.06722	**1.91441 ± 0.069**	1.95983 ± 0.04012	**1.93705 ± 0.06083**
	SC	0.20382 ± 0.01777	**0.20817 ± 0.01887**	0.1938 ± 0.00344	**0.2002 ± 0.01529**
	SD	1.64826 ± 0.0283	**1.64815 ± 0.02832**	1.66252 ± 0.03457	**1.6553 ± 0.0324**
	RT	0.94748 ± 0.09631	**0.92626 ± 0.10121**	0.99837 ± 0.03155	**0.96234 ± 0.08407**
LRDB2	DB	1.52436 ± 0.42002	**1.47528 ± 0.37001**	1.68744 ± 0.41411	**1.63843 ± 0.38459**
	SC	0.33395 ± 0.10959	**0.34382 ± 0.10014**	0.28805 ± 0.1093	**0.29857 ± 0.1039**
	SD	2.84239 ± 0.63412	**2.76349 ± 0.55905**	3.09039 ± 0.6263	**3.01145 ± 0.58457**
	RT	0.69375 ± 0.47179	**0.62644 ± 0.41411**	0.86921 ± 0.46626	**0.80205 ± 0.43891**
LRDB3	DB	2.94263 ± 0.12489	**2.93696 ± 0.12788**	3.0114 ± 0.22726	**2.97292 ± 0.17355**
	SC	0.10263 ± 0.00735	**0.10265 ± 0.00686**	0.10017 ± 0.01064	**0.10042 ± 0.00948**
	SD	2.32874 ± 0.07231	**2.32833 ± 0.07218**	2.30435 ± 0.15084	**2.30381 ± 0.09601**
	RT	2.20612 ± 0.20542	**2.19989 ± 0.20849**	2.31537 ± 0.38257	**2.25861 ± 0.28156**
LRDB4	DB	1.94253 ± 0.24398	**1.89247 ± 0.18771**	1.99679 ± 0.22767	**1.89885 ± 0.27487**
	SC	0.21184 ± 0.03164	**0.21962 ± 0.02426**	0.2048 ± 0.02778	**0.2205 ± 0.02898**
	SD	2.02541 ± 0.18196	**2.01309 ± 0.158**	2.1079 ± 0.18926	**1.94093 ± 0.13944**
	RT	1.00316 ± 0.2711	**0.93779 ± 0.20157**	1.05623 ± 0.25264	**0.94993 ± 0.31184**

Table 3– Comparative average analysis of efficiency CT measure ($\boldsymbol{means \pm std}$) over 10 trials

DS	KM	**SRSE-KM**	KM++	**SRSE-KM++**
LRDB1	8.83084 ± 2.35755	**2.31235 ± 0.24165**	12.60682 ± 1.69874	**1.92721 ± 0.57942**
LRDB2	7.7521 ± 2.25847	**1.60213 ± 0.09076**	**16.63186 ± 3.9364**	**1.58328 ± 0.64967**
LRDB3	11.26206 ± 2.51383	**1.34438 ± 0.03677**	24.22219 ± 4.92122	**1.50246 ± 0.05333**
LRDB4	6.29301 ± 3.01334	**3.34744 ± 5.10325**	43.21801 ± 6.65378	**1.96903 ± 0.42415**

Figure 2-5 depicts a comparative analysis of clustering objective and efficiency-related measures, with the resulting values ordered ascending to identify minimum to maximum values. The WSSE clustering objective for the KM and SRSE-KM algorithms is depicted in Figure 2, whereas the WSSE clustering objective for the KM++ and SRSE-KM++ algorithms is depicted in Figure 3. The minimum WSSE result shows that the proposed sampling plan consistently achieves the excellence WSSE in each trial on the experimental loan risk data sets. The observation of Figures 2-3 indicates that the SRSE based clustering algorithm achieves better compaction and separation of the cluster with the clustering objective.

Figure 4 demonstrates the computing time efficiency measurements for the KM and SRSE-KM algorithms, while Figure 5 reveals the computing time efficiency measurements for the KM++ and SRSE-KM++ algorithms. The proposed sampling plan minimizes the computation cost, iterations, number of distances, and data comparisons in each trial on the experimental risk data sets, implying that the proposed sampling plan minimizes the computation cost, iterations, number of distances, and data comparisons in each trial on the experimental risk data sets. Figures 4-5 illustrate that the SRSE-based clustering algorithm outperforms the KM and KM++ algorithms in terms of speed and resilience.

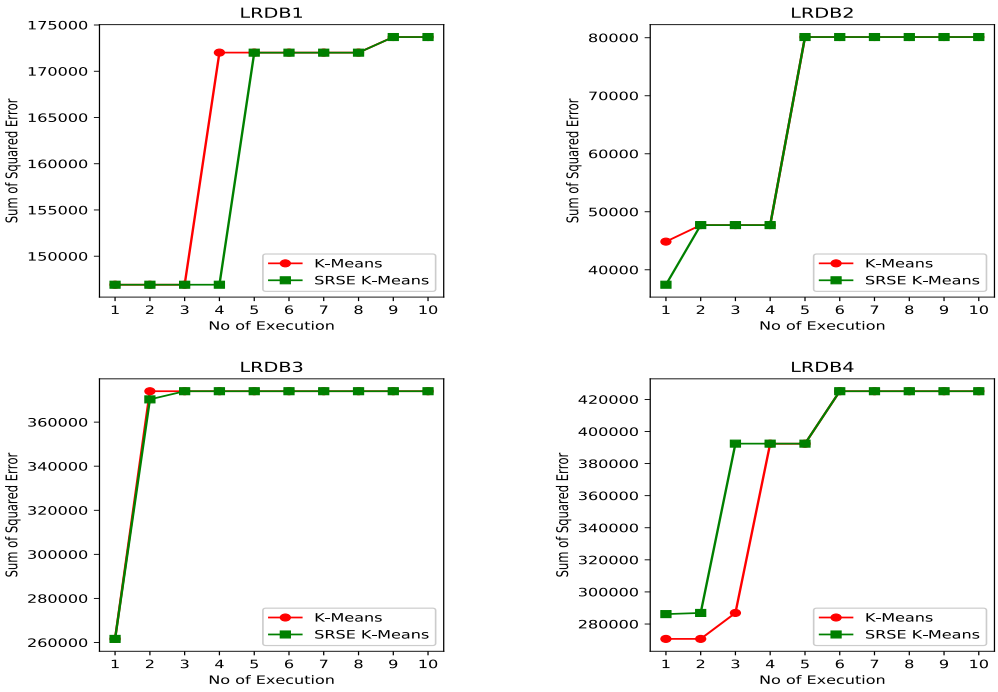

Figure 2 Analysis of total-WSSE between KM and SRSE-KM on each trial

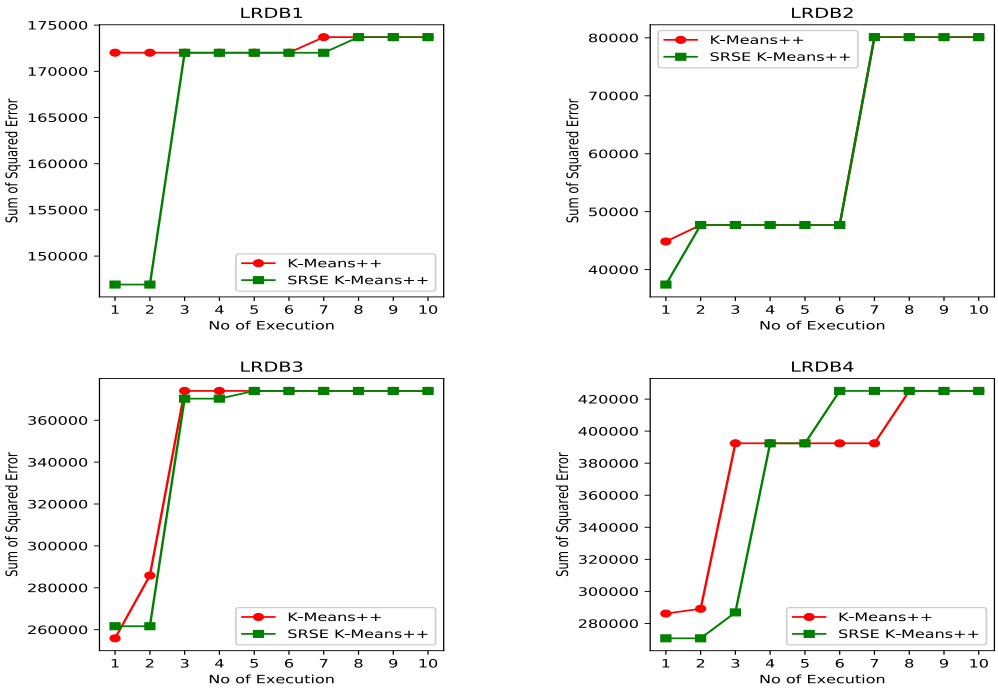

Figure 3 Analysis of total-WSSE between KM++ and SRSE-KM++ on each trial

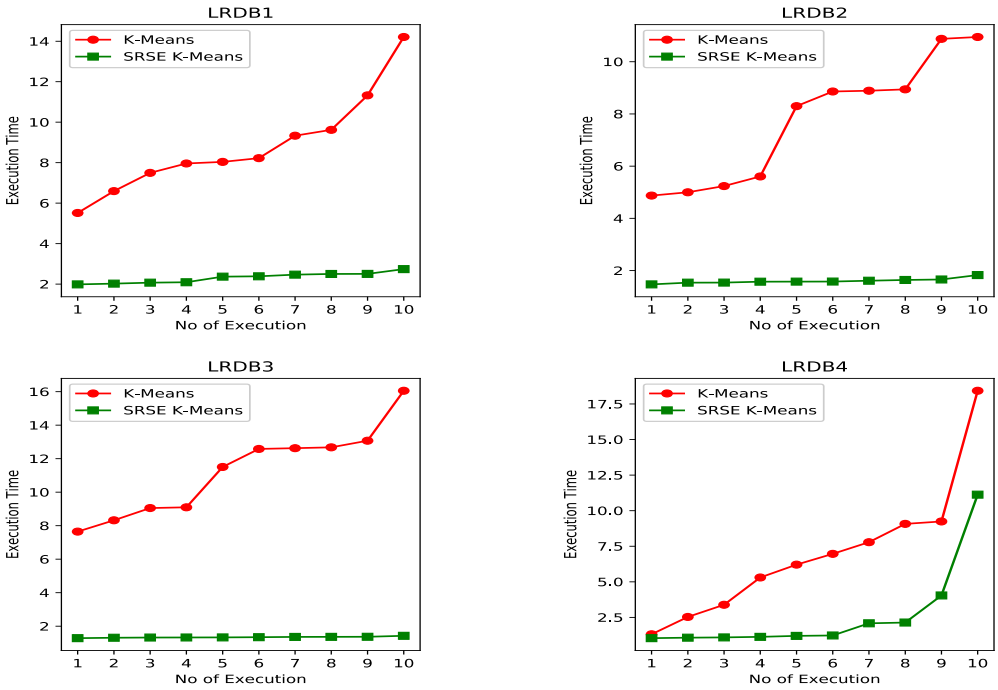

Figure 4 Analysis of computing time between KM and SRSE-KM on each trial

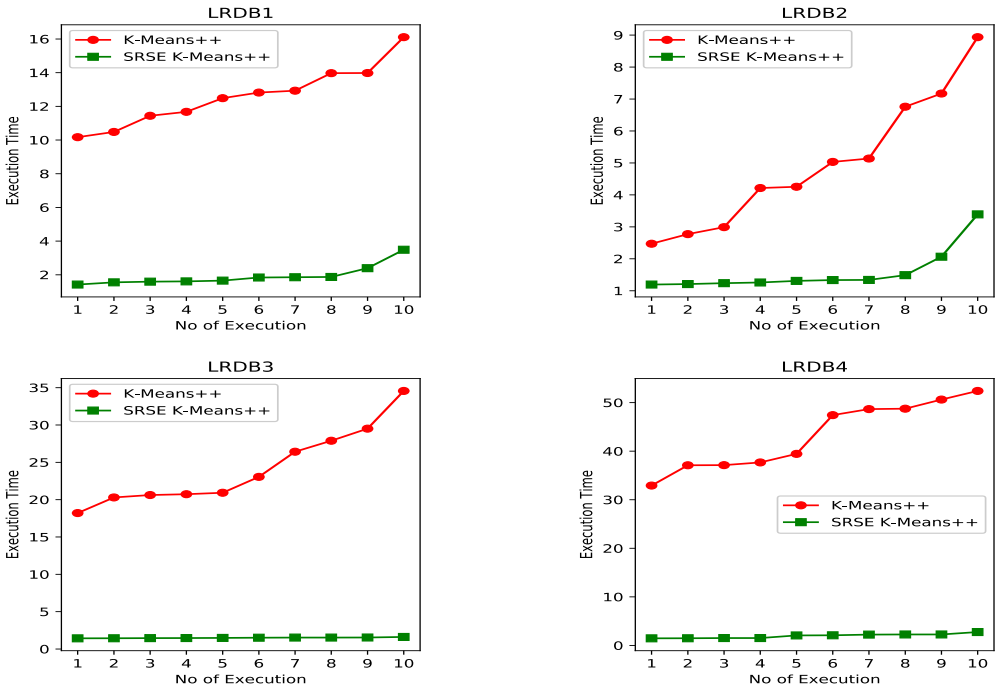

Figure 5 Analysis of computing time between KM++ and SRSE-KM++ on each trial

The proposed sampling-based clustering algorithm improves cluster quality and clustering objective while reducing data and distance comparisons with execution times, as shown in Table 2-3 and Figure 2-5. The presented sampling-based clustering algorithm also outperforms previous KM and KM++ algorithms in terms of speed and scalability on loan risk-based big data. On loan risk-based big data, the SRSE-based clustering algorithm eliminates the worst-case scenario of the KM and KM++ algorithms. The analysis shows that the proposed SRSE-KM and SRSE-KM++ algorithms are more robust for big data clustering than the KM and KM++ algorithms.

V. Conclusion

This study presents, a stratified remainder linear systematic sampling extension (SRSE) based clustering approach for loan risk analysis on big data using the KM and KM++ clustering algorithms. The proposed clustering SRSE-KM and SRSE-KM++ algorithms employ five stages to reduce computing time while improving cluster quality. The first step is to sort the entire dataset in order to create a stratification using the maximum variance attribute approach. The second stage determines the total sample size, sample interval, and reminder sample values in order to calculate the total number of data items and sample size in stratum. The third stage divides the data points into strata and uses a liner systematic sampling procedure to extract sample data from each stratum. The fourth stage clusters the sample data according to the selected clustering algorithm. The final stage uses a centroid-based sample extension approach to merge the sample data results to an un-sample data unit. The final results demonstrate that the unknown loan uncertainty of risk belongs to one cluster and non-risky data belongs to another cluster. Experiment results show that the SRSE-based clustering algorithm never degrades cluster performance and achieves better cluster compaction, separation, variance, density, computing cost, and execution time than classical clustering algorithms. The proposed SRSE-KM algorithm reduces average computing time by up to 75.25% when compared to KM, and the SRSE-KM++ algorithm reduces average computing time by up to 92.78% when compared to KM++. Despite the fact that the SRSE-based clustering algorithm significantly reduces clustering time, but it suffers local optima issues due to randomization. The study's further scope is to open up to resolve local optima concerns on multiple machine-based technologies such as Hadoop and Spark via other internal and external validation indexes using various loan risk-related data sets.

References

[1] Lozada N, Arias-Pérez J, Perdomo-Charry G (2019) Big data analytics capability and co-innovation: An empirical study. Heliyon 5:10. https://doi.org/10.1016/j.heliyon.2019.e02541

[2] Hariri RH, Fredericks EM, Bowers KM (2019) Uncertainty in big data analytics: survey, opportunities, and challenges. J Big Data 6:44. https://doi.org/10.1186/s40537-019-0206-3

[3] Tabesh P, Mousavidin E, Hasani S (2019) Implementing big data strategies: A managerial perspective. Bus Horiz 62:347–358. https://doi.org/10.1016/j.bushor.2019.02.001

[4] Elgendy N, Elragal A (2014) Big Data Analytics: A Literature Review Paper. In: Perner P (ed) ICDM 2014, LNAI 8557. Springer International Publishing Switzerland, pp 214–227

[5] Sivarajah U, Kamal MM, Irani Z, Weerakkody V (2017) Critical analysis of big data challenges and analytical methods. J Bus Res 70:263–286. https://doi.org/10.1016/j.jbusres.2016.08.001

[6] Gandomi A, Haider M (2015) Beyond the hype: big data concepts methods and analytics. Int J Inf Manage 35:137–144. https://doi.org/10.1016/j.ijinfomgt.2014.10.007

[7] Pandey KK, Shukla D (2019) Challenges of big data to big data mining with their processing framework. In: 2018 8th International Conference on Communication Systems and Network Technologies (CSNT). IEEE, pp 89–94

[8] Kacfah Emani C, Cullot N, Nicolle C (2015) Understandable big data: a survey. Comput Sci Rev 17:70–81. https://doi.org/10.1016/j.cosrev.2015.05.002

[9] Moharm K (2019) State of the art in big data applications in microgrid: A review. Adv Eng Informatics 42:. https://doi.org/10.1016/j.aei.2019.100945

[10] Khondoker MR (2018) Big data clustering. In: Wiley StatsRef: Statistics Reference Online.

John Wiley & Sons, Ltd, Chichester, UK, pp 1–10

[11] Pandove D, Goel S, Rani R (2018) Systematic review of clustering high-dimensional and large datasets. ACM Trans Knowl Discov Data 12:1–68. https://doi.org/10.1145/3132088

[12] Xie H, Zhang L, Lim CP, et al (2019) Improving K-means clustering with enhanced Firefly Algorithms. Appl Soft Comput J 84:105763. https://doi.org/10.1016/j.asoc.2019.105763

[13] Kara ME (2018) Supplier Risk Assessment Based on Best-Worst Method and K-Means Clustering: A Case Study. Sustainability 10:1066. https://doi.org/10.3390/su10041066

14. Mandelli D, Yilmaz A, Aldemir T, et al (2013) Scenario clustering and dynamic probabilistic risk assessment. Reliab Eng Syst Saf 115:146–160. https://doi.org/10.1016/j.ress.2013.02.013

[15] Marle F, Vidal L, Bocquet J (2013) Interactions-based risk clustering methodologies and algorithms for complex project management. Int J Prod Econ 142:225–234. https://doi.org/10.1016/j.ijpe.2010.11.022

[16] Kou G, Peng Y, Wang G (2014) Evaluation of clustering algorithms for financial risk analysis using MCDM methods. Inf Sci (Ny) 275:1–12. https://doi.org/10.1016/j.ins.2014.02.137

[17] Wang X, Frattini P, Stead D, et al (2020) Dynamic rockfall risk analysis. Eng Geol 272:105622. https://doi.org/10.1016/j.enggeo.2020.105622

[18] Caicedo PE, Rengifo CF, Rodriguez LE, et al (2020) Dataset for gait analysis and assessment of fall risk for older adults. Data Br 33:106550. https://doi.org/10.1016/j.dib.2020.106550

[19] 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 Comput 2:267–279. https://doi.org/10.1109/TETC.2014.2330519

[20] Jain AK (2010) Data clustering: 50 years beyond k-means. Pattern Recognit Lett 31:651–666. https://doi.org/10.1016/j.patrec.2009.09.011

[21] Wang X, He Y (2016) Learning from Uncertainty for Big Data: Future Analytical Challenges and Strategies. IEEE Syst Man, Cybern Mag 2:26–31. https://doi.org/10.1109/msmc.2016.2557479

[22] Zhao X, Liang J, Dang C (2019) A stratified sampling based clustering algorithm for large-scale data. Knowledge-Based Syst 163:416–428. https://doi.org/10.1016/j.knosys.2018.09.007

[23] Liu Z, Zhang A (2020) A Survey on Sampling and Profiling over Big Data (Technical Report). 1–17

[24] Mahmud MS, Huang JZ, Salloum S, et al (2020) A survey of data partitioning and sampling methods to support big data analysis. Big Data Min Anal 3:85–101. https://doi.org/10.26599/BDMA.2019.9020015

[25] Umarani V, Punithavalli M (2011) Analysis of the progressive sampling-based approach using real life datasets. Open Comput Sci 1:221–242. https://doi.org/10.2478/s13537-011-0016-y

[26] Chen B, Haas P, Scheuermann P (2002) A new two-phase sampling based algorithm for discovering association rules. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Digital Library, pp 462–468

[27] Furht B, Villanustre F (2016) Big Data Technologies and Applications. Springer International Publishing, Cham

[28] Ramasubramanian K, Singh A (2016) Sampling and Resampling Techniques. In: Machine Learning Using R. pp 67–127

[29] Haas PJ (2016) Data-Stream Sampling: Basic Techniques and Results. Springer-Verlag Berlin Heidelberg

[30] Kim JK, Wang Z (2019) Sampling Techniques for Big Data Analysis. Int Stat Rev 87:S177–S191. https://doi.org/10.1111/insr.12290

[31] Brus DJ (2019) Sampling for digital soil mapping: A tutorial supported by R scripts. Geoderma 338:464–480. https://doi.org/10.1016/j.geoderma.2018.07.036

[32] Aune-Lundberg L, Strand G (2014) Comparison of variance estimation methods for use with two-dimensional systematic sampling of land use/land cover data. Environ Model Softw 61:87–97. https://doi.org/10.1016/j.envsoft.2014.07.001

[33] Satyanarayana A (2014) Intelligent sampling for big data using bootstrap sampling and chebyshev inequality. Can Conf Electr Comput Eng. https://doi.org/10.1109/CCECE.2014.6901029

[34] Buddhakulsomsiri J, Parthanadee P (2008) Stratified random sampling for estimating billing accuracy in health care systems. Health Care Manag Sci 11:41–54. https://doi.org/10.1007/s10729-007-9023-x

[35] da Silva A, Chiky R, Hébrail G (2012) A clustering approach for sampling data streams in sensor networks. Knowl Inf Syst 32:1–23. https://doi.org/10.1007/s10115-011-0448-7

[36] Rajasekaran S, Saha S (2013) A novel deterministic sampling technique to speedup clustering algorithms. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 8347 LNAI:34–46. https://doi.org/10.1007/978-3-642-53917-6_4

[37] Jaiswal R, Kumar A, Sen S (2014) A Simple D 2 -Sampling Based PTAS for k -Means. 22–46. https://doi.org/10.1007/s00453-013-9833-9

[38] Xu Z, Wu Z, Cao J, Xuan H (2015) Scaling Information-Theoretic Text Clustering: A Sampling-based Approximate Method. In: Proceedings - 2014 2nd International Conference on Advanced Cloud and Big Data, CBD 2014. pp 18–25

[39] Luchi D, Santos W, Rodrigues A, Varejao FM (2015) Genetic Sampling k -means for Clustering Large Data Sets. In: CIARP 2015, LNCS 9423. pp 691–698

[40] Jing L, Tian K, Huang JZ (2015) Stratified feature sampling method for ensemble clustering of high dimensional data. Pattern Recognit 48:3688–3702. https://doi.org/10.1016/j.patcog.2015.05.006

[41] Li M, Li D, Shen S, et al (2016) DSS: A Scalable and Efficient Stratified Sampling Algorithm for Large-Scale Datasets. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp 133–146

[42] Zhan Q (2017) Improved spectral clustering based on Nyström method. Multimed Tools Appl 76:20149–20165. https://doi.org/10.1007/s11042-017-4566-4

[43] Aloise D, Contardo C (2018) A sampling-based exact algorithm for the solution of the minimax diameter clustering problem. J Glob Optim 71:613–630. https://doi.org/10.1007/s10898-018-0634-1

[44] Reddy KG, Khan MGM (2019) Optimal stratification in stratified designs using weibull-distributed auxiliary information. Commun Stat - Theory Methods 48:3136–3152. https://doi.org/10.1080/03610926.2018.1473609

[45] Saini M, Kumar A (2018) Ratio estimators using stratified random sampling and stratified ranked set sampling. Life Cycle Reliab Saf Eng. https://doi.org/10.1007/s41872-018-0046-8

[46] Li F, Qian Y, Wang J, et al (2019) Clustering ensemble based on sample's stability. Artif Intell 273:37–55. https://doi.org/10.1016/j.artint.2018.12.007

[47] Goshu NN, Kassa SM (2020) A Systematic Sampling Evolutionary (SSE) Method for Stochastic Bilevel Programming Prob-. Comput Oper Res 104942. https://doi.org/10.1016/j.cor.2020.104942

[48] Rajendra Prasad K, Mohammed M, Narasimha Prasad L V., Anguraj DK (2021) An efficient sampling-based visualization technique for big data clustering with crisp partitions. Distrib Parallel Databases 39:813–832. https://doi.org/10.1007/s10619-021-07324-3

[49] Nguyen TD, Shih MH, Srivastava D, et al (2021) Stratified random sampling from streaming and stored data. Springer US

[50] Larson L, Larson P, Johnson DE (2019) Differences in Stubble Height Estimates Resulting from Systematic and Random Sample Designs. Rangel Ecol Manag 72:586–589.

https://doi.org/10.1016/j.rama.2019.03.007

[51] Pandey kamlesh kumar, Shukla D (2020) Stratified Sampling-Based Data Reduction and Categorization Model for Big Data Mining. In: Gupta JC, Kumar BM, Sharma H, Agarwal B (eds) Communication and Intelligent Systems

[52] Pandey KK, Shukla D (2019) Optimized sampling strategy for big data mining through stratified sampling. Int J Sci Technol Res 8:3696–3702

[53] Xiao Y, Yu J (2012) Partitive clustering (k -means family). Wiley Interdiscip Rev Data Min Knowl Discov 2:209–225. https://doi.org/10.1002/widm.1049

[54] Rice JA (2007) Mathematical statistics and metastatistical analysis, Third Edit. Thomson Higher Education

[55] Singh S (2003) Advanced sampling theory with applications, volume 1. Springer Netherlands, Dordrecht

[56] Chang H-J, Huang K-C (2000) Remainder Linear Systematic Sampling. Indian J Stat Ser B 62:249–56

[57] Mostafa SA, Ahmad IA (2018) Recent developments in systematic sampling: a review. J Stat Theory Pract 12:290–310. https://doi.org/10.1080/15598608.2017.1353456

[58] Cochran WG (1962) Samling Techniques. Asia Publishing House, Bombay

[59] Arthur D, Vassilvitskii S (2007) K-means++: The advantages of careful seeding. In: SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. ACM Digital Library, pp 1027–1035

[60] Fränti P, Sieranoja S (2019) How much can k-means be improved by using better initialization and repeats? Pattern Recognit 93:95–112. https://doi.org/10.1016/j.patcog.2019.04.014

[61] HajKacem MA Ben, N'Cir C-E Ben, Essoussi N (2019) Overview of scalable partitional methods for big data clustering. In: Nasraoui O, N'Cir C-E Ben (eds) Clustering Methods for Big Data Analytics, Unsupervised and Semi-Supervised Learning. Springer Nature, Switzerland, pp 1–23

[62] Aggarwal CC, Reddy CK (2014) Data custering algorithms and applications. CRC Press, Boca Raton, United States

[63] Sid R, H TR (1999) Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation. In: 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99). Narosa Publishing House, pp 137–143

[64] Peña J., Lozano J., Larrañaga P (1999) An empirical comparison of four initialization methods for the k-means algorithm. Pattern Recognit Lett 20:1027–1040. https://doi.org/10.1016/S0167-8655(99)00069-0

[65] Celebi ME, Kingravi HA, Vela PA (2013) A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst Appl 40:200–210. https://doi.org/10.1016/j.eswa.2012.07.021

[66] Zahra S, Ghazanfar MA, Khalid A, et al (2015) Novel centroid selection approaches for k-means-clustering based recommender systems. Inf Sci (Ny) 320:156–189. https://doi.org/10.1016/j.ins.2015.03.062