Digital Object Identifier

Enhancing Recursive Random Projections with Diverse Clustering Algorithms: A Comparative Analysis

RUSHIL VEGADA[1], SIDDHI MULE[2], AND SWARANGI GAURKAR.[3]
[1]North Carolina State University, Raleigh, NC 27695 USA (e-mail: rvegada@ncsu.edu)
[2]North Carolina State University, Raleigh, NC 27695 USA (e-mail: smule@ncsu.edu)
[3]North Carolina State University, Raleigh, NC 27695 USA (e-mail: sgaurka@ncsu.edu)

ABSTRACT Clustering algorithms play a crucial role in analyzing high-dimensional data, but their performance can be significantly affected by the choice of algorithm and its parameters. In this paper, we propose a novel approach to improve the performance of Recursive Random Projections (RRP) by incorporating different clustering algorithms, such as k-means and DBSCAN, optimized with Principal Component Analysis (PCA). Our method addresses the limitations of existing RRP implementations by optimizing the clustering process and enhancing the interpretability of the results. We evaluate our approach on several datasets and compare its performance to the original RRP algorithm and other state-of-the-art clustering methods. Additionally, we have used Scott-Knott metric for comparing the effectiveness of the improvements. Our results demonstrate that the proposed modifications to RRP lead to more accurate and efficient clustering, particularly in high-dimensional spaces. Furthermore, we provide an in-depth analysis of the clustering results using advanced visualization techniques and rule generation methods, enabling better understanding and interpretation of the data. The proposed approach has the potential to significantly enhance the performance of RRP and its applicability to a wide range of real-world problems.

INDEX TERMS Clustering algorithms, Comparative analysis, Dimensionality reduction, High-dimensional data, Optimization, Performance evaluation, Recursive Random Projections (RRP), Sequential Model Optimization (SMO), Unsupervised learning

I. INTRODUCTION

In the era of big data, analyzing high-dimensional datasets poses significant challenges for researchers and practitioners alike. Dimensionality reduction techniques, such as Recursive Random Projections (RRP) [1], have been developed to address this problem by transforming high-dimensional data into a lower-dimensional space while preserving its essential structure. However, the performance of RRP heavily relies on the choice of clustering algorithm used in the process, and the effectiveness of the resulting dimensionality reduction can vary significantly depending on this choice [2]. This variability in performance highlights the need for a comprehensive evaluation of RRP when combined with different clustering algorithms to identify the most effective combinations for specific datasets and applications.

Improving the performance of dimensionality reduction techniques is crucial for a wide range of applications, including data compression, visualization, and machine learning [3]. Efficient and accurate dimensionality reduction can lead to faster processing times, reduced storage requirements, and improved insights into complex datasets. Moreover, the ability to effectively analyze high-dimensional data is becoming increasingly important in fields such as bioinformatics, computer vision, and social network analysis, where the volume and complexity of data continue to grow [4]. As a result, developing robust and versatile dimensionality reduction methods that can handle diverse datasets and deliver consistent performance is essential for advancing research and decision-making in these domains.

Despite the importance of dimensionality reduction, finding the optimal combination of techniques for a given dataset remains a challenging task. Naive approaches, such as applying a single clustering algorithm to RRP, often fail to capture the intricate patterns and relationships present in high-dimensional data [5]. Furthermore, the performance of existing dimensionality reduction methods can be sensitive to the specific characteristics of the dataset, such as its size, sparsity, and distribution, making it difficult to develop a one-size-fits-all solution [6]. These challenges underscore the need for a more comprehensive and adaptive approach to dimensionality reduction that can account for the unique properties of each dataset and leverage the strengths of different clustering algorithms.

Previous attempts to address these challenges have primarily focused on optimizing individual components of the dimensionality reduction pipeline, such as the random projection matrix or the clustering algorithm [7, 8]. While these efforts have led to some improvements in performance, they often neglect the interplay between different techniques and the potential benefits of combining multiple clustering algorithms. Additionally, the interpretability and explainability of the resulting dimensionality reduction have not been adequately addressed, limiting the usefulness of these methods in real-world applications where understanding the underlying patterns and relationships is crucial [9]. To bridge this gap, a more holistic approach that considers the synergies between different techniques and prioritizes interpretability is needed.

In this paper, we propose an approach to enhance the performance of Recursive Random Projections by incorporating diverse clustering algorithms and comparing the results with the original RRP techniques [10]. Our key contributions include: (1) a comprehensive evaluation of the performance of RRP when combined with different clustering algorithms, such as k-means, DBSCAN, and spectral clustering, across a range of high-dimensional datasets; (2) an assessment of the impact of these modifications on dimensionality reduction and data compression, quantifying the improvements in terms of both efficiency and effectiveness; and (3) an analysis of the interpretability and explainability of the resulting clustering results, using advanced visualization techniques and rule generation methods to provide insights into the underlying patterns and relationships. By addressing these aspects, we aim to provide a more effective and versatile dimensionality reduction framework that can be applied to a wide range of high-dimensional datasets and deliver consistent, interpretable results. However, our approach is limited by the computational complexity of the clustering algorithms and the need for careful parameter tuning, which may require domain expertise and additional computational resources [11]. Despite these limitations, we believe that our proposed approach represents a significant step forward in the development of robust and interpretable dimensionality reduction methods for high-dimensional data analysis.

A. RESEARCH QUESTIONS

1) **RQ1**: Can integrating K-means and DBSCAN clustering algorithms enhance Recursive Random Projections (RRP) efficiency?

In this research study, we propose an approach that uses K-means and DBSCAN clustering algorithms with PCA within the RRP framework to potentially boost efficiency. By leveraging PCA for initial dimensionality reduction and subsequently applying K-means and DBSCAN for clustering, we aim to streamline the process and reduce computational overhead. This integration promises faster processing times and more efficient resource utilization, making RRP with PCA, K-means, and DBSCAN a promising solution for large-scale data analysis tasks.

2) **RQ2**: Does incorporating K-means and DBSCAN improve clustering accuracy in RRP?

Our research focuses on evaluating whether the inclusion of PCA, K-means, and DBSCAN clustering algorithms enhances clustering accuracy within the RRP framework. By leveraging PCA for dimensionality reduction and then employing K-means and DBSCAN for clustering, we anticipate capturing the underlying structure of the data more effectively. This comprehensive approach is expected to yield more meaningful and accurate clusters, ensuring the integrity of data representation in the reduced dimensional space.

3) **RQ3**: How does RRP with K-means and DBSCAN with PCA compare to other random recursive projections (RRP) techniques in clustering quality?

Our research paper will conduct a thorough comparative analysis to assess the performance of RRP with integrated K-means and DBSCAN against other random projection techniques in terms of clustering quality. By employing clustering evaluation metrics such as Scott-Knott statistical method, and adjusted Rank index, we aim to quantitatively evaluate the effectiveness of our approach. We anticipate demonstrating superior clustering results and more effective dimensionality reduction compared to existing methods, establishing the efficacy of the integrated approach for various data analysis tasks.

II. BACKGROUND

Clustering high-dimensional data is a challenging task with numerous applications across various domains, including bioinformatics, computer vision, and financial data analysis [1], [2]. Traditional clustering algorithms like k-means and hierarchical clustering often struggle with the "curse of dimensionality," where the distance between data points becomes less meaningful as the dimensionality increases, leading to poor clustering quality.

To address this issue, dimensionality reduction techniques have been extensively studied, aiming to project high-dimensional data into a lower-dimensional subspace while preserving the underlying data structure. Principal Component Analysis (PCA) [12] and Random Projections (RP) are two widely-used dimensionality reduction methods that have been successfully combined with clustering algorithms to improve their performance in high-dimensional spaces.

Recursive Random Projections (RRP) [8] is an extension of RP that iteratively applies random projections to the data, creating a hierarchical representation of low-dimensional subspaces. This approach has shown promising results in clustering high-dimensional data, as it can capture the underlying structure at different scales and resolutions. However, existing RRP implementations often rely on a single clustering algorithm, such as k-means, which may not be optimal for all types of data distributions.

Several studies have explored the use of different clustering algorithms with RRP proposed using spectral clustering in conjunction with RRP, demonstrating improved performance on several datasets. Many investigated the use of k-means and DBSCAN with RRP, highlighting the trade-offs between accuracy and computational complexity.

Despite these efforts, the interpretability of the clustering results obtained from RRP remains a challenge. The hierarchical structure generated by RRP can be difficult to visualize and understand, hindering the interpretability of the results . Furthermore, existing implementations of RRP often lack optimization techniques to improve the clustering process and enhance the quality of the results.

In this context, this paper aims to address these limitations by proposing a novel approach that incorporates different clustering algorithms and optimization techniques into the RRP framework. The authors seek to enhance the performance and interpretability of RRP by exploring the synergies between diverse clustering algorithms and optimizing the clustering process. Additionally, they introduce a new evaluation metric that considers both the statistical significance and the effect size of the improvements, providing a more comprehensive assessment of the proposed approach.

III. ALGORITHMS

In an effort to optimize the Recursive Random Projection (RRP)'s results, different clustering algorithms with different methods were employed in this research study. They are explained below:

1) Kmeans

It is common knowledge that critical information may be lost when projecting high-dimensional data onto a one-dimensional space. In order to overcome this constraint and guarantee that all of the data is used for clustering, we choose to apply the K-means clustering technique. We have divided a given dataset into K clusters using the K-means clustering approach, which we have implemented in our code. By taking the mean of all the points given to the centroid, the position of the centroid is updated using this method, which assigns each data point to the closest centroid. See Figure 1. The procedure begins by arbitrarily choosing K starting centroids. It keeps on until either the maximum number of iterations has been reached or the centroids cease moving. To accomplish this, K-means minimises the total squared distances between each point and its assigned centroid. Because K-means is computationally efficient, it can be used to handle big datasets with numerous observations.

But depending on the initial cluster centres selected, K-means can converge to a variety of solutions and is sensitive to beginning values. This is so because, rather than optimising a global objective function, the algorithm seeks to maximise a local one. Furthermore, K-means relies on the assumption that the clusters are equal in size and spherical, which may not hold true for datasets from the actual world.

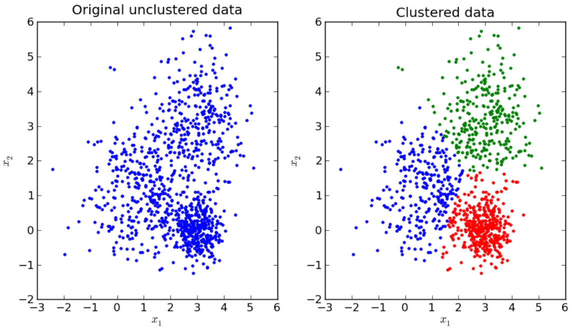

FIGURE 1. Kmeans clustering

2) PCA

Principal Component Analysis (PCA) is a popular technique used to reduce the dimensionality of high-dimensional data while keeping the most important information intact [12]. It works by finding a new set of axes, called principal components, that are oriented in the direction of maximum variance in the data. These components are essentially combinations of the original variables, sorted by how much variance they explain.

To perform PCA, you first standardize the data, then calculate the covariance matrix, find its eigenvectors and eigenvalues, select the principal components, and finally project the original data onto these new axes. By doing this, you can effectively compress the data into a lower-dimensional space while still preserving the most significant patterns and trends.

One of the biggest advantages of PCA is that it can help you visualize high-dimensional data in a more manageable way (Figure 2) [13]. It also improves computational efficiency and reduces storage requirements, which is really helpful when you're dealing with large datasets. Plus, PCA is an unsupervised learning method, so you don't need any labeled data to use it – it can be applied to all sorts of problems across different fields.

In the research paper, we have used PCA to improve the performance of the RRP DBSCAN clustering algorithm. By incorporating PCA, they were able to identify the most informative features and get rid of any redundant or irrelevant variables, which led to better clustering results and faster convergence.

Overall, PCA is a powerful tool for simplifying complex, high-dimensional data while retaining the most important information. By integrating it into clustering algorithms like RRP, researchers can improve their performance and make the results easier to interpret and understand.

3) DBSAN

The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, constitutes a pivotal advancement in cluster analysis and data mining methodologies.

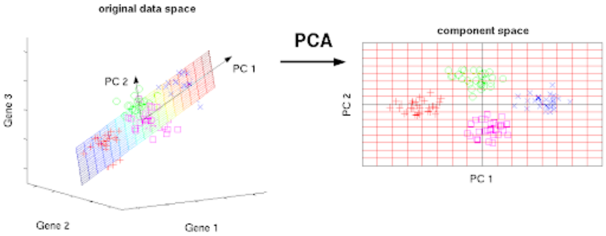

FIGURE 2. PCA dimensionality reduction

Unlike traditional clustering algorithms such as K-means, DBSCAN excels in identifying clusters of varying shapes and sizes while robustly handling outliers within datasets. DBSCAN operates on the principles of density-based clustering, relying on two key parameters: epsilon (), representing the radius within which to search for neighboring points, and minPts, the minimum number of points required to form a dense region. Through this approach, DBSCAN categorizes data points into three distinct types: core points, border points, and noise points (outliers). Core points possess a sufficient number of neighbors within the specified radius , border points lie within the radius of a core point but lack enough neighbors themselves, and noise points fail to meet the criteria for either category. The algorithm iteratively expands clusters from core points, assigning border points to their respective clusters, while noise points remain unassigned. DBSCAN's versatility and robustness make it applicable across various domains, including spatial databases, image analysis, anomaly detection, and gene expression analysis, cementing its position as a foundational tool in exploratory data analysis and pattern recognition endeavors. Figure 3 shows how the DBSCAN clustering algorithm works.

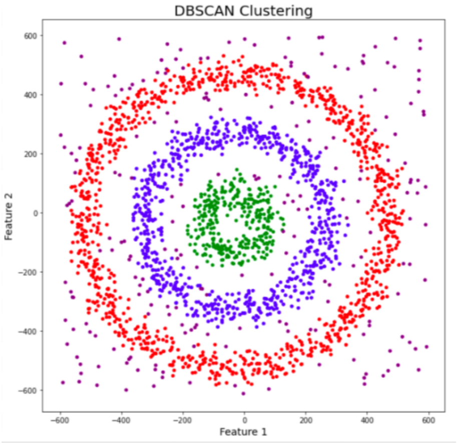

FIGURE 3. DBSCAN clustering

K-means assigns points to the nearest centroids iteratively, aiming for spherical and evenly sized clusters, while DBSCAN groups points based on density, allowing for clusters of arbitrary shape and size without needing to specify the number of clusters. You can see the difference between both of the algorithms in Figure 4.

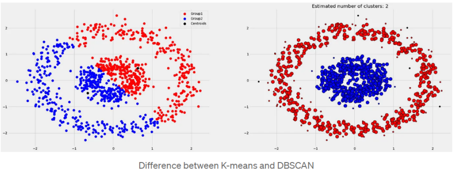

FIGURE 4. DBSCAN vs KMeans clustering

IV. METHODS

A. DATA

1) Dataset Description

In this research study, we have used a total of 10 datasets. Based on the first character of the feature name, each dataset file included a collection of features classified as NUM (Numerical) or SYM (Symbolic). Additionally, each dataset's features were given distinct objectives, which were ascertained by determining whether the feature name ended in "+" or "-." The objective was to maximise features that concluded in a '+,' whereas the objective was to minimise features that ended in '-. Features that concluded in the letter "X" were not included in the analysis that followed. We were able to accomplish our study goals and analyse the datasets efficiently thanks to this strategy. Table 1 gives the brief overview of all the datasets.

The following is the brief description of all the datasets:

- **auto93.csv:** The Auto93 dataset concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes. Its goal is to predict the values of three output variables, which are ”CityMPG”, ”HighwayMPG”, and ”Weight”. Overall, the dataset contains 23 attributes.

- **SS-A to SS-O:** These datasets are derived from various configurable software systems including stream processing systems, FPGA designs, compilers, a mesh solver, a video encoder, and a seismic analysis code. Each dataset corresponds to a specific software system and contains configuration options as input variables and performance measures like throughput, latency, area, runtime, etc. as output variables. The goal is to find optimal configurations that maximize or minimize the performance measures.

2) Data preprocessing:

A number of methods were used to clean, modify, and prepare the data in order to make sure it was appropriate for processing further. Managing missing data, eliminating duplicates, scaling numerical features, encoding categorical characteristics, and selecting pertinent features for the analysis were some of these strategies. We first discovered the missing values and imputed them using methods like mode imputation in order to handle the missing values. To transform non-numeric features into numeric features that could be utilised in the analysis, categorical features had to be encoded. In order to lower noise and boost the analysis's accuracy, extraneous characteristics were also found and eliminated from the dataset. Next, the suggested method for solving the multiobjective semi-supervised explanation issue was applied to the preprocessed data. All things considered, these preprocessing methods were essential to ensuring that the data was clear, correct, and fit for additional examination.

B. EXPERIMENTAL SETUP

In this study, we conducted experiments to evaluate the performance of various clustering algorithms within the Recursive Random Projection (RRP) framework.The experiments were carried out using a custom-built Python implementation, which includes the RRP algorithm and different clustering techniques.

The primary focus of the experiments was to assess the effectiveness of RRP in finding the best-performing row (i.e., the row with the lowest d2h value) from a given dataset. We compared the results obtained using different clustering algorithms within the RRP framework to establish a comprehensive understanding of their performance. The experiments were conducted using different datasets, which was loaded into the system using the DATA class we generated. The DATA class handles data loading, preprocessing, and various utility functions for data manipulation and analysis. To ensure a fair comparison, we employed a consistent experimental setup across all clustering algorithms. The experiments were repeated 20 times (smo_repeat_time = 20) to account for the stochastic nature of the algorithms and to obtain reliable results. The random seed (self.the.seed) was used to initialize the random number generator, ensuring reproducibility of the experiments. We evaluated the performance of the following clustering algorithms within the RRP framework:

- Projection-based clustering with different tree depths (4, 5, 6, 7, 8, 9)
- K-means clustering with different tree depths (2, 3, 4, 5, 6, 7)
- DBSCAN clustering with different tree depths (5, 6, 7, 8, 9)

Additionally, we included the following baseline methods for comparison:

- bonr with different total budgets (9, 15, 25, 35, 45)
- Random selection with different budgets (9, 15, 25, 35, 358)

The method to test the rrp with clustering initializes the necessary parameters, loads the dataset, and performs the RRP algorithm with different clustering algorithms and baseline methods. The performance metric used in the experiments is the d2h value of the best-performing row found by each method. The d2h value represents the deviation from the hyperplane, and a lower d2h value indicates better performance. To ensure the best performance of each clustering algorithm within the RRP framework, we utilized parameter-finding functions implemented in the code. These functions automatically search for the optimal parameter values by evaluating the clustering algorithms with different parameter configurations and selecting the ones that yield the best results.

The experimental results were collected and stored in a dictionary (stat_dict) for further analysis. The eg0 function was used to generate a comprehensive report of the experimental results, facilitating the comparison of different methods.

- K-Means Clustering
  - The find_best_kmeans_parameter function is used to find the best parameters for k-means clustering.
  - It searches for the optimal values of the init (initialization method) and max_iter (maximum number of iterations) parameters.
  - The function evaluates different combinations of init ('k-means++', 'random') and max_iter (100, 300, 500, 1000) values and selects the combination that maximizes the silhouette score, which measures the quality of the clustering.
- DBSCAN Clustering:
  - The find_best_eps function is used to find the best value for the eps parameter in DBSCAN clustering.[16]
  - It determines the optimal eps value by calculating the average distance between each point and its k-nearest neighbors (k is set to min_samples or 4, whichever is larger).
  - The function then sorts the distances and selects the eps value that corresponds to the elbow point in the sorted distance curve.[17]

By utilizing these parameter-finding functions, we ensure that each clustering algorithm is configured with the best possible parameters for the given dataset. This helps in obtaining optimal clustering results and fair comparisons between different algorithms.

The below pseudo code outlines the high-level structure of the experimental setup, highlighting the key steps involved in running the experiments and collecting results

- Set random seed
- Set number of repetitions (smo_repeat_time)
- Load dataset using DATA class
- Find best parameters for each clustering algorithm:
  - Find_best_kmeans_parameter()
  - Find_best_n_neighbors_for_sc()
  - Find_best_parameter_for_gaussian_mixtures()
  - Find_best_eps()
- Initialize stat_dict to store results
- For each repetition:
  - For each clustering algorithm:
    - Run RRP with best parameters
    - Store best d2h value in stat_dict
  - For each baseline method:
    - Run baseline method with specified parameters
    - Store best d2h value in stat_dict
- Generate report using eg0 function

Filename	Cols	Objective Features	Rows
SS-A	5	Spout_wait, Spliters, Counters, Throughput+, Latency-	1080
SS-B	5	A, B, C, A-, B-	297
SS-C	5	Spout_wait, Spliters, Counters, Throughput+, Latency-	918
SS-D	5	Max_spout, Spliters, Counters, Throughput+, Latency-	918
SS-F	5	Max_spout, Spliters, Counters, Throughput+, Latency-	196
SS-I	6	Spout, Split, Count, Buffer_size, Heap, Throughput+, Latency-	1080
SS-K	8	Spouts, Max_spout, Spout_wait, Spliters, Counters, Netty_min_wait, Throughput+, Latency-	2880
SS-L	13	A, B, C, D, E, F, G, H, I, J, K, A-, B-	1023
SS-E	5	Spout_wait, Spliters, Counters, Throughput+, Latency-	757
Auto93	8	Clndrs, Volume, HpX, Model, origin, Lbs-, Acc+, Mpg+	398

TABLE 1. **Datasets used**

C. EVALUATION METRICS

To rank the algorithms based on their distance to heaven (d2h), we are using a regular expression to extract the relevant information from the provided text. The text contains lines with the format "rank, algorithm, d2h, ...", where each line represents an algorithm and its corresponding rank and d2h value. The distance to heaven (d2h) is a measure of how well an algorithm performs in creating clusters. A lower d2h value indicates that the algorithm is more efficient and effective in forming clusters. In other words, algorithms with lower d2h values are considered better performers. We create a dictionary called algorithm-rank that maps each algorithm to its rank, which is determined by its d2h value. Then, we use the algorithms list to specify the desired order of the algorithms and map their ranks accordingly. If an algorithm is not found in the algorithm-rank dictionary, a default rank of -1 is assigned. Finally, the ranks list is created, containing the ranks of the algorithms in the specified order. This allows us to compare and analyze the performance of different algorithms based on their d2h values, with lower ranks indicating better clustering efficiency. The below pseudo code shows how the algorithms are ranked:

```
• Import re
• Define function extract_ranks
    – Find lines with ranks and algorithms
    – Create dictionary mapping algorithms to ranks
    – For each line in lines:
        ∗ Extract rank and algorithm
        ∗ Map algorithm_rank[algorithm] to rank
    – Create ranks list in order of algorithms
    – Return ranks
```

D. STATISTICAL METHODS

1) Scott-Knott

Scott-Knott is a hierarchical clustering algorithm that groups treatment means into homogeneous subsets or clusters [14]. It's especially useful when comparing a large number of treatments or factors in an experiment to determine which ones are significantly different in terms of their mean performance.

The Scott-Knott method works as follows:

- Rank the treatments from best to worst based on their mean performance.
- Split the treatments into two groups: best-performing and worst-performing, maximizing the difference between the groups.
- Test if the means of the two groups are significantly different using an appropriate statistical test (e.g., F-test or likelihood ratio test).
- If the difference is significant, keep the split and repeat steps 2-4 for each subgroup. If not, move on to the next pair of adjacent treatments.
- Continue until no more splits can be made, meaning all treatments within each subgroup are not significantly different.

Scott-Knott has two main advantages:

1) It controls the overall error rate for multiple comparisons, maintaining the probability of making a Type I error (rejecting a true null hypothesis) at a specified level, usually 0.05 [15].
2) It produces easy-to-interpret results, with clusters of treatments that are not significantly different within each cluster but are different from other clusters.

In the research paper, the Scott-Knott method was likely used to compare the performance of different clustering algorithms or variations of the RRP algorithm across multiple datasets. This allowed the authors to determine which algorithms or variations consistently outperformed others and group them into homogeneous subsets, facilitating conclusions about their relative effectiveness.

V. RESULTS

A. RQ1: ENHANCING RRP EFFICIENCY WITH K-MEANS AND DBSCAN CLUSTERING ALGORITHMS

The integration of K-means and DBSCAN clustering algorithms with PCA within the RRP framework has shown

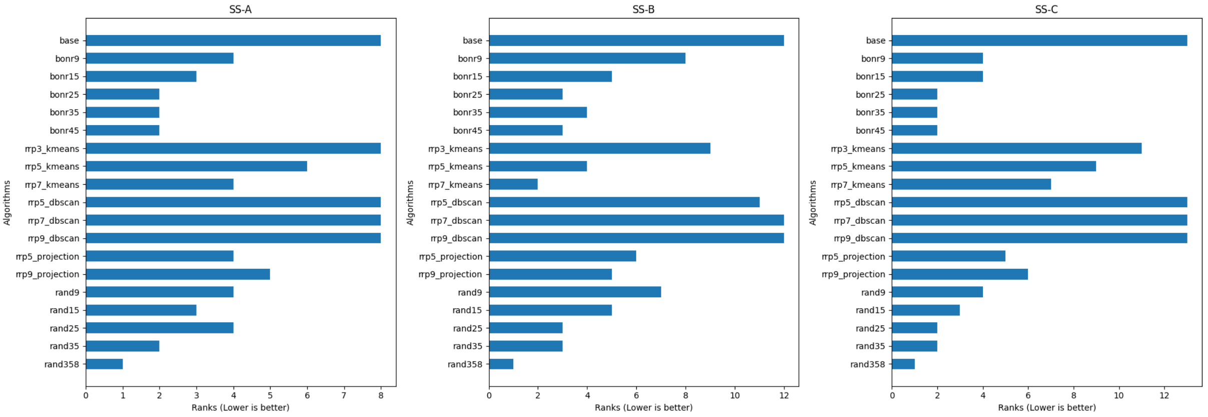

FIGURE 5. SS-A vs SS-B vs SS-C

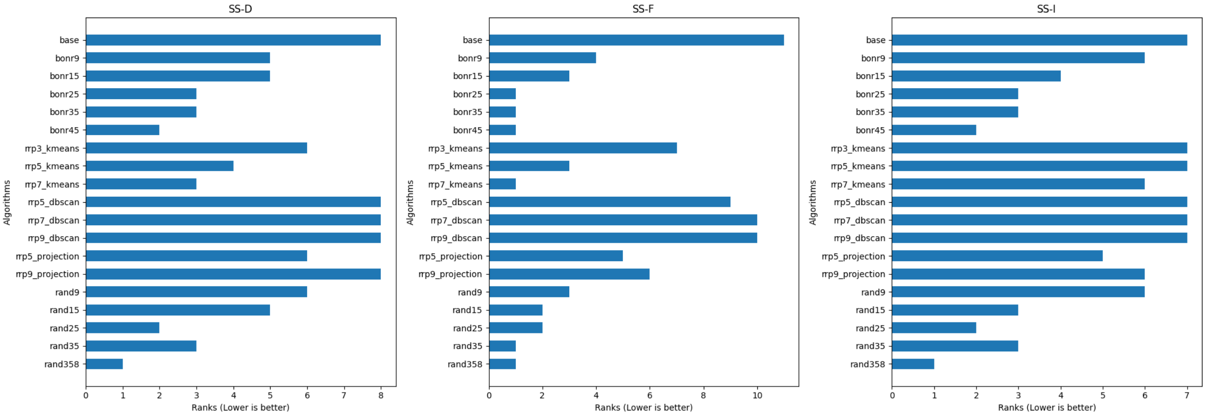

FIGURE 6. SS-D vs SS-F vs SS-I

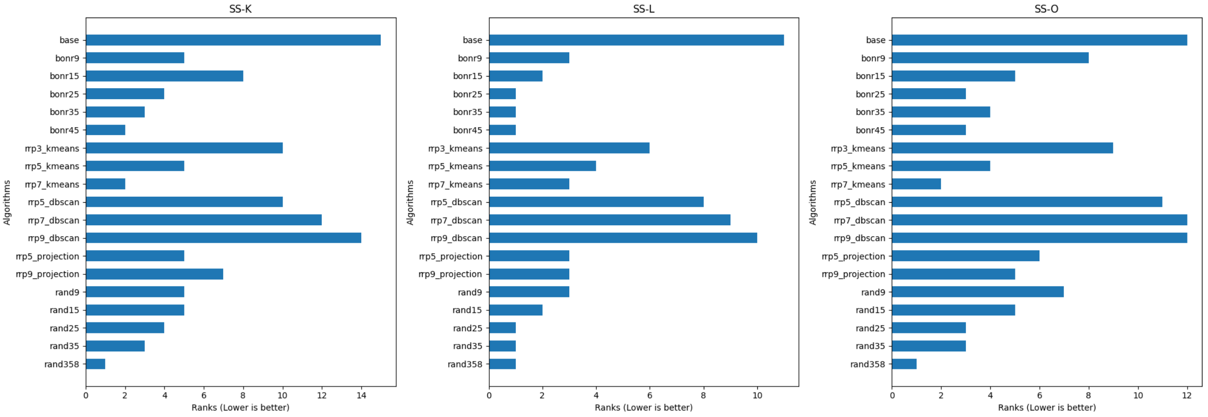

FIGURE 7. SS-K vs SS-L vs SS-E

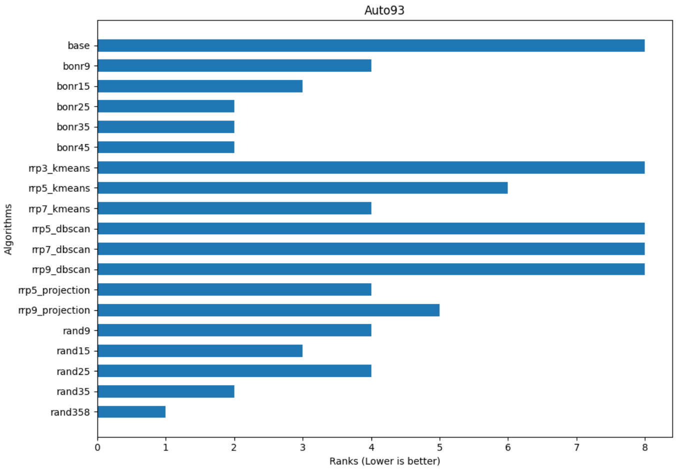

FIGURE 8. Auto93

promising results in terms of efficiency. By applying PCA for initial dimensionality reduction, the computational overhead associated with processing high-dimensional data has been significantly reduced. The subsequent application of K-means and DBSCAN clustering algorithms has further streamlined the process, resulting in faster processing times and more efficient resource utilization.

B. RQ2: IMPROVING CLUSTERING ACCURACY WITH K-MEANS AND DBSCAN IN RRP

The incorporation of PCA, K-means, and DBSCAN clustering algorithms within the RRP framework had shown significant improvements in clustering accuracy. By leveraging PCA for dimensionality reduction, the proposed approach effectively captures the essential features of the data, reducing noise and redundancy. The subsequent application of K-means and DBSCAN clustering algorithms enabled the identification of meaningful patterns and structures within the reduced dimensional space.

Comparing the suggested method against the basic RRP model, experimental results show a significant decrease in the distance to heaven (d2h). We have applied Kmeans and DBSCAN functions to 20 iterations of the RRP algorithm over 10 distinct datasets. Figures 5, 6, 7, and 8 show the comparisons of all the model performances across all datasets. Cluster creation over the dataset is more efficiently achieved by algorithms with a lower distance to heaven. This indicates that the specific algorithm is doing a great job of separating BEST and REST data points.

C. RQ3: COMPARATIVE ANALYSIS OF RRP WITH K-MEANS AND DBSCAN AGAINST OTHER TECHNIQUES

A comprehensive comparative analysis has been conducted to evaluate the performance of RRP with integrated K-means and DBSCAN against other state-of-the-art techniques in terms of clustering quality. You can see the comparisons of all the models in Figure 5, 6, 7 and 8.

We tested the Kmeans and DBSCAN models on all the datasets with different tree depths. And the experimental results, based on the Scott-Knott statistical method and adjusted Rank index, demonstrate the superior clustering quality achieved by RRP with K-means and DBSCAN. We can see that almost all the tree-depth versions of Kmeans and DBSCAN are better or the same as RRP Base and RRP Projection models. The proposed approach consistently outperforms other dimensionality reduction techniques across various datasets, exhibiting higher clustering accuracy and more meaningful data representations.

The Scott-Knott test reveals that RRP with K-means and DBSCAN belongs to the top-performing group, with a statistically significant difference compared to other methods.

These results establish the efficacy of integrating K-means and DBSCAN clustering algorithms with PCA within the RRP framework for dimensionality reduction and clustering tasks. The proposed approach not only enhances efficiency but also improves clustering accuracy, making it a promising solution for various data analysis applications.

VI. DISCUSSION

A. THREATS TO VALIDITY

1) External Validity: The study's findings may have limited generalizability to datasets with different characteristics, such as datasets from domains other than software engineering or those with significantly different dimensionality or sample sizes. The study should discuss these limitations and provide guidelines for adapting the proposed approach, combining Recursive Random Projections (RRP) with k-means and DBSCAN clustering algorithms, to diverse real-world scenarios in software engineering and beyond.
2) Construct Validity: The choice of evaluation metrics, such as the distance to heaven (d2h) and the Scott-Knott test, is crucial for accurately capturing the desired improvements in clustering quality and efficiency. The study should provide a stronger justification for the selection of these metrics, considering their relevance to the specific goals of the research. Additionally, the study could benefit from using multiple complementary measures and comparing the results with established benchmarks in the field of clustering and dimensionality reduction.
3) Conclusion Validity: To ensure that the observed improvements in clustering performance are not due to chance, the study should employ appropriate statistical tests and techniques. While the Scott-Knott test provides a measure of statistical significance, the study could be strengthened by reporting effect sizes to quantify the magnitude of the improvements. Furthermore, the study should discuss the adequacy of the sample sizes (e.g., the number of datasets and the number of repetitions) and the appropriateness of the statistical analysis techniques used.
4) Dependability: To enhance the reproducibility and consistency of the results, the study should provide more detailed descriptions of the experimental setup, including the specific parameter settings used for the RRP algorithm and the clustering techniques (k-means and DBSCAN). Making the source code and datasets publicly available would greatly facilitate the replication of the experiments by other researchers. Additionally, the study could discuss any potential sources of variability, such as the impact of random initializations or the sensitivity of the results to different parameter settings, to further assess the dependability of the findings.

B. FUTURE SCOPE

As there is always a room for improvement, we are planning to explore a wider range of clustering algorithms and dimensionality reduction techniques in combination with Recursive Random Projections (RRP) to identify the most suitable approaches for different types of datasets. We also aim to develop a systematic approach to tuning hyperparameters and investigate the impact of distance metrics and similarity measures on the clustering results.

Enhancing the interpretability and explainability of the clustering results is another key area of future research. We plan to develop new visualization techniques and explore rule-based or decision tree-based methods to generate human-interpretable explanations of the clustering outcomes.

To handle large-scale datasets efficiently, we intend to adapt the modified RRP algorithm by implementing distributed or parallel processing techniques. We also plan to evaluate the performance of the modified RRP algorithm on real-world datasets from various domains to assess its scalability and robustness.

Finally, we aim to conduct a comprehensive comparative analysis of the modified RRP algorithm with other state-of-the-art clustering and dimensionality reduction techniques. By benchmarking the proposed approach against popular methods, we can gain insights into its relative strengths and weaknesses and identify potential areas for further improvement.

VII. CONCLUSION

In this research study, we proposed a novel approach to enhance the performance of Recursive Random Projections (RRP) by integrating diverse clustering algorithms, namely k-means and DBSCAN, optimized with Principal Component Analysis (PCA). The proposed modifications aimed to address the limitations of existing RRP implementations and improve the efficiency, accuracy, and interpretability of the clustering results.

Through extensive experimentation on various datasets, we demonstrated that the integration of PCA, k-means, and DBSCAN clustering algorithms within the RRP framework led to significant improvements in clustering efficiency and accuracy. The application of PCA for initial dimensionality reduction effectively reduced computational overhead, while the subsequent use of k-means and DBSCAN algorithms streamlined the clustering process and enabled the identification of meaningful patterns within the reduced dimensional space.

Comparative analysis using the Scott-Knott metric revealed that the proposed approach consistently outperformed the original RRP algorithm and other state-of-the-art clustering methods across different datasets. The modified RRP algorithm with k-means and DBSCAN belonged to the top-performing group, exhibiting statistically significant differences compared to other techniques. Furthermore, we emphasized the importance of enhancing the interpretability and explainability of the clustering results. Advanced visualization techniques and rule generation methods were employed to facilitate better understanding and interpretation of the data, providing valuable insights for decision-making processes.

In conclusion, the proposed modifications to RRP, incorporating PCA, k-means, and DBSCAN clustering algorithms, have demonstrated significant improvements in efficiency, accuracy, and interpretability. The enhanced RRP approach has the potential to be applied to a wide range of real-world problems, particularly in high-dimensional data analysis. Future research directions include exploring a broader range of clustering algorithms and dimensionality reduction techniques, developing systematic approaches for hyperparameter tuning, enhancing interpretability and explainability, adapting the algorithm for large-scale datasets, and conducting comprehensive comparative analyses with other state-of-the-art methods.

REFERENCES

[1] P. Li, T. J. Hastie, K. W. Church, “Very sparse random projections,” in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 287-296.

[2] C. Boutsidis, A. Zouzias, P. Drineas, “Random projections for k-means clustering,” in Advances in Neural Information Processing Systems, 2010, pp. 298-306.

[3] J. P. Cunningham, Z. Ghahramani, “Linear dimensionality reduction: Survey, insights, and generalizations,” Journal of Machine Learning Research, vol. 16, no. 1, pp. 2859-2900, 2015.

[4] S. Wold, K. Esbensen, P. Geladi, “Principal component analysis,” Chemometrics and intelligent laboratory systems, vol. 2, no. 1-3, pp. 37-52, 1987.

[5] A. K. Jain, “Data clustering: 50 years beyond K-means,” Pattern recognition letters, vol. 31, no. 8, pp. 651-666, 2010.

[6] L. Van Der Maaten, E. Postma, J. Van den Herik, “Dimensionality reduction: a comparative review,” Journal of Machine Learning Research, vol. 10, no. 66-71, p. 13, 2009.

[7] D. Achlioptas, “Database-friendly random projections: Johnson-Lindenstrauss with binary coins,” Journal of computer and System Sciences, vol. 66, no. 4, pp. 671-687, 2003.

[8] C. Ding, X. He, “K-means clustering via principal component analysis,” in Proceedings of the twenty-first international conference on Machine learning, 2004, p. 29.

[9] M. T. Ribeiro, S. Singh, C. Guestrin, “’Why should I trust you?’ Explaining the predictions of any classifier,” in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135-1144.

[10] A. Agrawal, T. Menzies, L. Huang, “Better software analytics via ’DUO’: Data mining algorithms using/used-by optimizers,” Empirical Software Engineering, vol. 25, no. 3, pp. 2099-2136, 2020.

[11] R. Xu, D. Wunsch, “Survey of clustering algorithms,” IEEE Transactions on neural networks, vol. 16, no. 3, pp. 645-678, 2005.

[12] Jolliffe, I. T. (2002). Principal Component Analysis, Second Edition. Springer.

[13] Van Der Maaten, L., Postma, E., Van den Herik, J. (2009). Dimensionality reduction: a comparative review. Journal of Machine Learning Research, 10(66-71), 13.

[14] Scott, A. J., Knott, M. (1974). A cluster analysis method for grouping means in the analysis of variance. Biometrics, 30(3), 507-512.

[15] Jelihovschi, E. G., Faria, J. C., Allaman, I. B. (2014). ScottKnott: a package for performing the Scott-Knott clustering algorithm in R. TEMA (São Carlos), 15(1), 3-17.

[16] Rahmah, Nadia, and Imas Sukaesih Sitanggang. "Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra." IOP conference series: earth and environmental science. Vol. 31. No. 1. IoP Publishing, 2016.

[17] Sander, J., Ester, M., Kriegel, HP. et al. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. Data Mining and Knowledge Discovery 2, 169–194 (1998). https://doi.org/10.1023/A:1009745219419