SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I4P102

Volume 11 Issue 4, 9-23, April 2024
© 2024 Seventh Sense Research Group®

Original Article

Auto-Grouping Sedimentation Using Unsupervised Based Clustering Techniques

Radhika Surampudi[1], R. Kumudham[2]

[1,2]*Department of Electronics and Communication Engineering, Vels Institute of Science, Technology & Advanced Studies, Tamilnadu, India.*

[2]*Corresponding Author : kumudham.se@velsuniv.ac.in*

Received: 03 February 2024 Revised: 03 March 2024 Accepted: 01 April 2024 Published: 30 April 2024

Abstract **-** *A Machine Learning (ML) algorithm plays an important role in the prediction of inaccuracies in several fields, such as medicine, computer science, along underwater particle sedimentation. Hence, in this research work, the authors implemented various clustering methods for grouping the sediment particles such as mud, sand50, gravel50, rock 10 cm, rock 50cm, surface carbon, and nitrogen in the underwater sea automatically. This research focuses on the application of unsupervised machine learning, specifically clustering techniques, to automate the grouping of underwater sediment particles. The research highlights the utilization of K-means Clustering and BIRCH Clustering, introducing a novel contribution in the form of a Hybrid Clustering approach that integrates the benefits of both methods. This hybridization is designed to refine and enhance clustering results, presenting a promising solution for the automation of sediment analysis in underwater environments. To predict the performance of various unsupervised machine learning-based clustering algorithms, metrics like Calinski Harabasz, Silhouette Score, Mathew's Correlation Score, Davies Bouldin, Hamming loss, and Cohen Kappa score with n=7 are evaluated in underwater sediment particles grouping. Among several clustering techniques, the proposed hybrid approach outperforms in clustering of sediment articles based on the Silhouette score.*

Keywords **-** *Auto-grouping, Sedimentation, Unsupervised clustering methods, Hybrid clustering, Machine Learning.*

1. Introduction

Analysis of sedimentation in underwater environments holds a crucial role in comprehending the dynamics of aquatic ecosystems, significant for effective environmental monitoring. The accurate classification and grouping of sediment particles, encompassing a variety of elements such as mud, sand, gravel, rocks, and organic matter, contribute significantly to solving the complexities of underwater ecosystems.

Conventional methods of sediment analysis are often labour-intensive and time-consuming, necessitating advanced computational techniques to automate and enhance the precision of particle classification. Moreover, machine learning algorithms have emerged as powerful tools for predictive analysis and pattern recognition across various domains.

This research focuses on the field of unsupervised machine learning, specifically exploring the application of clustering techniques to automate the grouping of underwater sediment particles. The primary aim is to develop a model that not only expedites the sedimentation analysis process but also enhances the accuracy of classification, thereby contributing to a more comprehensive understanding of underwater environments.

The technique of permitting granules suspended in water to separate under the impact of gravity is called deposition. Sludge refers to the sedimentary fragments created because of dispersion within the water treatment process.

As the flow of water stops, the degraded particles, which are being carried by the stream, drop outside the aquatic environment and then into the bottom. The particles that make up a waterway's bottom, sides, and riverbank were carried through the stream of water from upstream in the watershed. There are various approaches to assessing sediment texture.

The initial method is grain size. According to the Wentworth scale, sediments can be categorized based on their particle size. The smallest sediments are made of clay with a grain diameter of less than .004 mm, while the largest are boulders with a grain diameter of at least 256 mm. Grain size, among diverse things, reflects the conditions of the sediment's deposition. Only the huge particles often settle in high-energy environments like strong waves or currents because the finer particles are carried away.

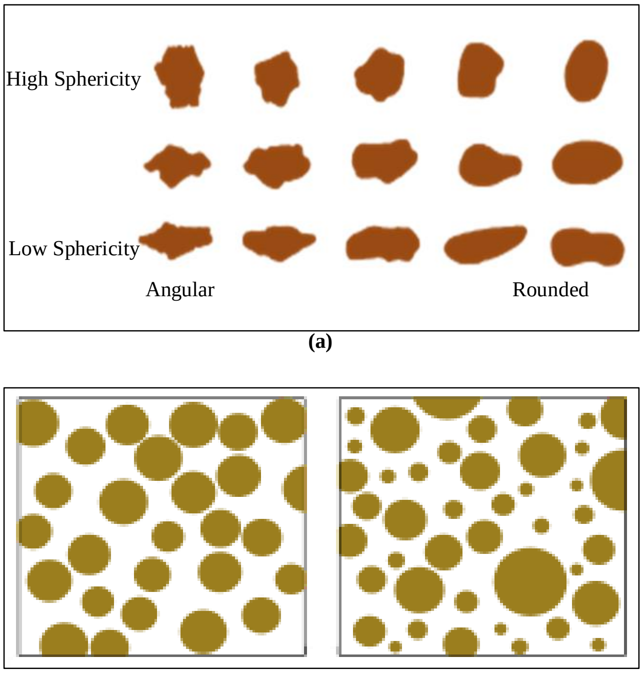


Fig. 1 (a) Well-Sorted sediment particles, and (b) Difference among angular and rounded.

Smaller particles will be able to separate out and produce finer sediments under lower energy circumstances. Figure 1 (a) discusses well-sorted sediment particles and Figure 1 (b) describes the differences in grain shape such as angular and rounded.

1.1. Problem Statement

The analysis of underwater sediment particles currently relies heavily on manual classification, a process known to be labour-intensive, time-consuming, and susceptible to human error. Although machine learning algorithms have demonstrated potential for automating this task, existing research primarily concentrates on individual clustering techniques, neglecting the exploration of hybrid approaches.

As a result, a gap exists in the investigation of hybrid clustering techniques specifically designed for underwater sediment analysis. This research proposes to address this gap by implementing and evaluating a novel Hybrid Clustering approach. This approach integrates the strengths of K-means Clustering and BIRCH Clustering methods, aiming to provide a more accurate and efficient solution for the automated grouping of underwater sediment particles.

The novelty of this research is the development and assessment of a Hybrid Clustering approach for automating underwater sediment particle grouping. This approach utilizes the combined strengths of K-means Clustering and BIRCH Clustering methodologies to achieve more refined and improved clustering outcomes. Based on this performance it provides a novel solution to the persistent challenge of manual categorization in underwater sediment analysis.

1.2. Research Objectives

The objectives of this research work are mentioned as follows:

- Implement the sedimentation dataset using unsupervised-based clustering methods which appropriate for grouping the similar sediment particles in one class and dissimilar sediment particles are arranged in another category.
- Several clustering-based techniques, such as k-means, mini-batch k-means, BIRCH, mean shift, OPTICS, and hybrid, are used to find the optimum solution for auto-grouping the sediment particles.
- The performance metrics such as Silhouette Score, Calinski Harabasz Score, Davies Bouldin, Mathew’s Correlation Score, Hamming Loss, and Cohen Kappa score are evaluated to predict clustering method performances in the detection of sediment particles.

The research employs two clustering techniques, K-means Clustering and BIRCH Clustering, and develops a novel hybrid clustering approach that integrates the advantages of both methods. K-means clustering, known for its simplicity and efficiency, forms the initial basis. In contrast, BIRCH clustering, with its hierarchical and summarization capabilities, is strategically integrated to refine and enhance the clustering results.

2. Related Works

The review of existing research works highlights the importance of studying underwater sediment particles for environmental monitoring and resource management. Machine learning techniques have been increasingly utilized to automate sediment classification using data from sonar imagery and sensor networks.

A critical review of the literature reveals a gap in the exploration of hybrid clustering methods specifically designed for automating the grouping of these particles. This review aims to integrate current knowledge in this field, emphasizing the need for novel approaches to improve the accuracy and efficiency of sediment analysis.

Nowadays, machine learning algorithms and their approaches provide solutions to all kinds of issues in every domain. Here, various literature works relevant to sediment classification underwater using machine learning and AI techniques were discussed. During the year 1994, Stewart et al. [1] introduced a back propagation neural network approach for classifying sediment particles like ridges, valleys and ponds using side scan sonar images on the mid-ocean ridge area database. Xiao Ming Qin et al. [2] found that sediments are categorized into small particles using a deep convolutional neural network with side scan sonar images of the underwater sea to attain large-span feature migration.

Abda et al. [3] determined that AI (a combination of particle swarm optimization, LSTM, and Random Forest along with an Artificial Neural Network) provides a better solution during flood disasters in the North Eastern Algerian River by replicating the suspended sediment load. Dillip et al. [4] developed a hybrid approach comprising machine learning algorithms and sensor networks for resolving the issues of sedimentation procedure during every month of torrential rain period.

Asadi et al. [5] employed several machine learning techniques like Artificial Neural Networks, SVM, Evolutionary SVM, and regression for identifying suspended sediment load in various canals submerged in countries like Iran, Gilan, and Lorestan convenience for terrain analysis aspects. Similarly, Nourani et al. [6] and Kermani et al. [7] identified sediment load using machine learning models and integrated machine learning methods as well. Moreover, Yilmaz et al. [8] calculated SSL using a spline-based regression technique, an optimization method was applied to obtain the optimal solution, Bee colony method-based artificial approach as well.

Aid Ahoul et al. [9] implemented a neural network-based Long Short-Term Memory approach for predicting Suspended Sediment Load, especially in Malaysia Johon River. Berthold et al. [10] applied deep deep-based convolutional neural network for categorizing sediments on the seafloor using a side scan sonar database with four groups, namely fine, sand, coarse and mixed sediments.

Huang et al. [11] proposed integrated machine learning techniques such as support vector and LSTM approach for finding suspended sediment Deliberation Lake during a hurricane in real-time situations. Awasthi et al. [12] used different approaches of acoustic signal processing for classifying sediments in sea beds based on materials uniqueness and parameters chosen from signals as well.

Cui et al. [13] classified ten kinds of sea sediments, such as sand, gravel, mud, etc, using a fuzzy ranking optimization approach on specific features. Based on classification accuracy, the performance of methods was evaluated in seafloor sediment classification.

Issam Mohamed et al. [14] used the Artificial Neural Network method for sediment classification on the Thames River database based on discharge of water, temperature of water and conductivity with electricity. Bhattacharya et al. [15] analyzed the database from the year 1992 to 1998 for sediment classification based on several factors. The accuracy found in this investigation is appropriate for identifying sediments in the seabed along with classification.

Liu et al. [16] developed machine learning-based techniques such as decision tree, random forest, and logistic regression for sediment identification in the seabed, especially in the ship shoal of Louisiana located in the United States. Based on certain six parameters, specifically bathymetry and backscatter features, the sediment classification with 58 subtypes was done.

Zhu et al. [17] calculate approximately the velocity of dregs settling underwater using machine learning techniques. Ojha et al. [18] suggested Bayesian neural network methods for sediment classification in the Bering Sea. Mitchell et al. [19] determined the rates of sediment particles present in seabed/seafloor on a database of the Baltic Sea using machine learning techniques. Mishra et al. [20] used the integration of machine learning and sensor networks for sediment classification. Qin et al. [21] determined the sediment particle classification on side scan sonar images like [22] in which how the classification of attacks performed. Siless et al. [23] explained several clustering approaches and their working principles appropriate for this research work on sediment particle grouping based on unsupervised machine techniques.

2.1. Research Gap Analyzed

A review of existing literature on underwater sediment classification reveals a vast array of machine learning and AI techniques applied across diverse datasets and scenarios. However, a closer examination highlights a significant research gap: the lack of exploration and evaluation of hybrid clustering techniques specifically designed for automating underwater sediment particle grouping. While numerous studies focus on individual machine learning models or techniques for sediment classification, only a few research works have used the potential advantages of combining multiple approaches, such as integrating clustering algorithms.

This research seeks to address this gap by proposing and evaluating a novel Hybrid Clustering approach. This approach aims to improve the accuracy and efficiency of underwater sediment analysis, ultimately contributing to advancements in the field of environmental monitoring and management.

3. Proposed Methodology

3.1. Dataset Description

In this research work, the sedimentation of underwater sea metadata has been gathered from specified resources for grouping certain sedimentation-based features. The dataset link is described as Table 1 in which the dataset has 1,11,111 rows with 20 features that are represented in columns wise.

Table 1. Sedimentation database link

S. No.	Dataset Link
1	https://pureportal.strath.ac.uk/en/datasets/data-for-a-synthetic-map-of-the-northwest-european-shelf-sediment

3.2. Removal of High Correlation Features

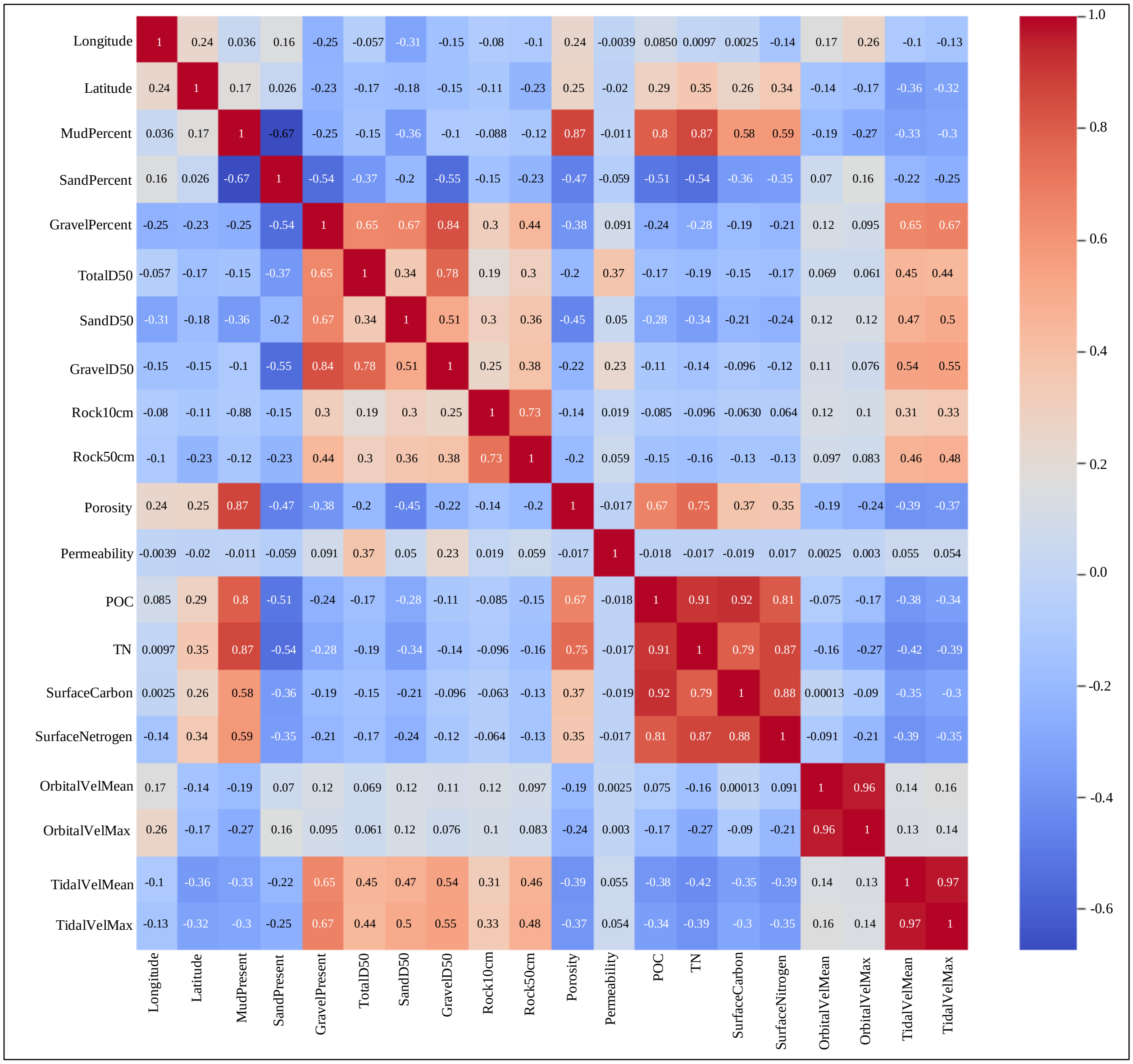
In fact, the availability of highly correlated features in the dataset causes more complexity to the algorithm; hence, the quantity of errors might be increased. To reduce the complexity of the program, the high correlation features are removed during this phase.

The following are the stages to drop out high correlation features: Importing the Python libraries, the dataset being loaded, building the correlation matrix, then choosing the upper triangular matrix, removing the features represented in highly correlated columns, and finally, the outcomes are analyzed.

The correlation matrix representation after the removal of highly correlated features of sedimentation particles is depicted in Figure 2. After eradicating the high correlation features, another similar features like OrbitalVelMax, OrbitalVelMean, TidalVelMax, and TidalVelMean are available in sedimentation metadata; hence, the authors keep one feature and discard the other feature. Finally, 1,11,111 rows, along with 18 features, are moved to the next phase.

3.3. Utilization of Sedimentation Features

In this phase, the features which are relevant to the sedimentation process have been chosen for grouping the underwater sediment particles. The features such as longitude, latitude, Tidal and orbital Vel Mean, permeability, totalD50, POC, TN, percent of sand and gravel are found to be irrelevant; hence, such parameters are detached. Therefore, the overall metadata is reduced into a similar number of rows with only seven features.



	Longitude	Latitude	MudPresent	SandPresent	GravelPresent	TotalD50	SandD50	GravelD50	Rock10cm	Rock50cm	Porosity	Permeability	POC	TN	SurfaceCarbon	SurfaceNitrogen	OrbitalVelMean	OrbitalVelMax	TidalVelMean	TidalVelMax
Longitude	1	0.24	0.036	0.16	-0.25	-0.057	-0.31	-0.15	-0.08	-0.1	0.24	-0.0039	0.0850	0.0097	0.0025	-0.14	0.17	0.26	-0.1	-0.13
Latitude	0.24	1	0.17	0.026	-0.23	-0.17	-0.18	-0.15	-0.11	-0.23	0.25	-0.02	0.29	0.35	0.26	0.34	-0.14	-0.17	-0.36	-0.32
MudPercent	0.036	0.17	1	-0.67	-0.25	-0.15	-0.36	-0.1	-0.088	-0.12	0.87	-0.011	0.8	0.87	0.58	0.59	-0.19	-0.27	-0.33	-0.3
SandPercent	0.16	0.026	-0.67	1	-0.54	-0.37	-0.2	-0.55	-0.15	-0.23	-0.47	-0.059	-0.51	-0.54	-0.36	-0.35	0.07	0.16	-0.22	-0.25
GravelPercent	-0.25	-0.23	-0.25	-0.54	1	0.65	0.67	0.84	0.3	0.44	-0.38	0.091	-0.24	-0.28	-0.19	-0.21	0.12	0.095	0.65	0.67
TotalD50	-0.057	-0.17	-0.15	-0.37	0.65	1	0.34	0.78	0.19	0.3	-0.2	0.37	-0.17	-0.19	-0.15	-0.17	0.069	0.061	0.45	0.44
SandD50	-0.31	-0.18	-0.36	-0.2	0.67	0.34	1	0.51	0.3	0.36	-0.45	0.05	-0.28	-0.34	-0.21	-0.24	0.12	0.12	0.47	0.5
GravelD50	-0.15	-0.15	-0.1	-0.55	0.84	0.78	0.51	1	0.25	0.38	-0.22	0.23	-0.11	-0.14	-0.096	-0.12	0.11	0.076	0.54	0.55
Rock10cm	-0.08	-0.11	-0.88	-0.15	0.3	0.19	0.3	0.25	1	0.73	-0.14	0.019	-0.085	-0.096	-0.0630	0.064	0.12	0.1	0.31	0.33
Rock50cm	-0.1	-0.23	-0.12	-0.23	0.44	0.3	0.36	0.38	0.73	1	-0.2	0.059	-0.15	-0.16	-0.13	-0.13	0.097	0.083	0.46	0.48
Porosity	0.24	0.25	0.87	-0.47	-0.38	-0.2	-0.45	-0.22	-0.14	-0.2	1	-0.017	0.67	0.75	0.37	0.35	-0.19	-0.24	-0.39	-0.37
Permeability	-0.0039	-0.02	-0.011	-0.059	0.091	0.37	0.05	0.23	0.019	0.059	-0.017	1	-0.018	-0.017	-0.019	0.017	0.0025	0.003	0.055	0.054
POC	0.085	0.29	0.8	-0.51	-0.24	-0.17	-0.28	-0.11	-0.085	-0.15	0.67	-0.018	1	0.91	0.92	0.81	-0.075	-0.17	-0.38	-0.34
TN	0.0097	0.35	0.87	-0.54	-0.28	-0.19	-0.34	-0.14	-0.096	-0.16	0.75	-0.017	0.91	1	0.79	0.87	-0.16	-0.27	-0.42	-0.39
SurfaceCarbon	0.0025	0.26	0.58	-0.36	-0.19	-0.15	-0.21	-0.096	-0.063	-0.13	0.37	-0.019	0.92	0.79	1	0.88	0.00013	-0.09	-0.35	-0.3
SurfaceNetrogen	-0.14	0.34	0.59	-0.35	-0.21	-0.17	-0.24	-0.12	-0.064	-0.13	0.35	-0.017	0.81	0.87	0.88	1	-0.091	-0.21	-0.39	-0.35
OrbitalVelMean	0.17	-0.14	-0.19	0.07	0.12	0.069	0.12	0.11	0.12	0.097	-0.19	0.0025	0.075	-0.16	0.00013	0.091	1	0.96	0.14	0.16
OrbitalVelMax	0.26	-0.17	-0.27	0.16	0.095	0.061	0.12	0.076	0.1	0.083	-0.24	0.003	-0.17	-0.27	-0.09	-0.21	0.96	1	0.13	0.14
TidalVelMean	-0.1	-0.36	-0.33	-0.22	0.65	0.45	0.47	0.54	0.31	0.46	-0.39	0.055	-0.38	-0.42	-0.35	-0.39	0.14	0.13	1	0.97
TidalVelMax	-0.13	-0.32	-0.3	-0.25	0.67	0.44	0.5	0.55	0.33	0.48	-0.37	0.054	-0.34	-0.39	-0.3	-0.35	0.16	0.14	0.97	1

Fig. 2 Correlation matrix for sedimentation database

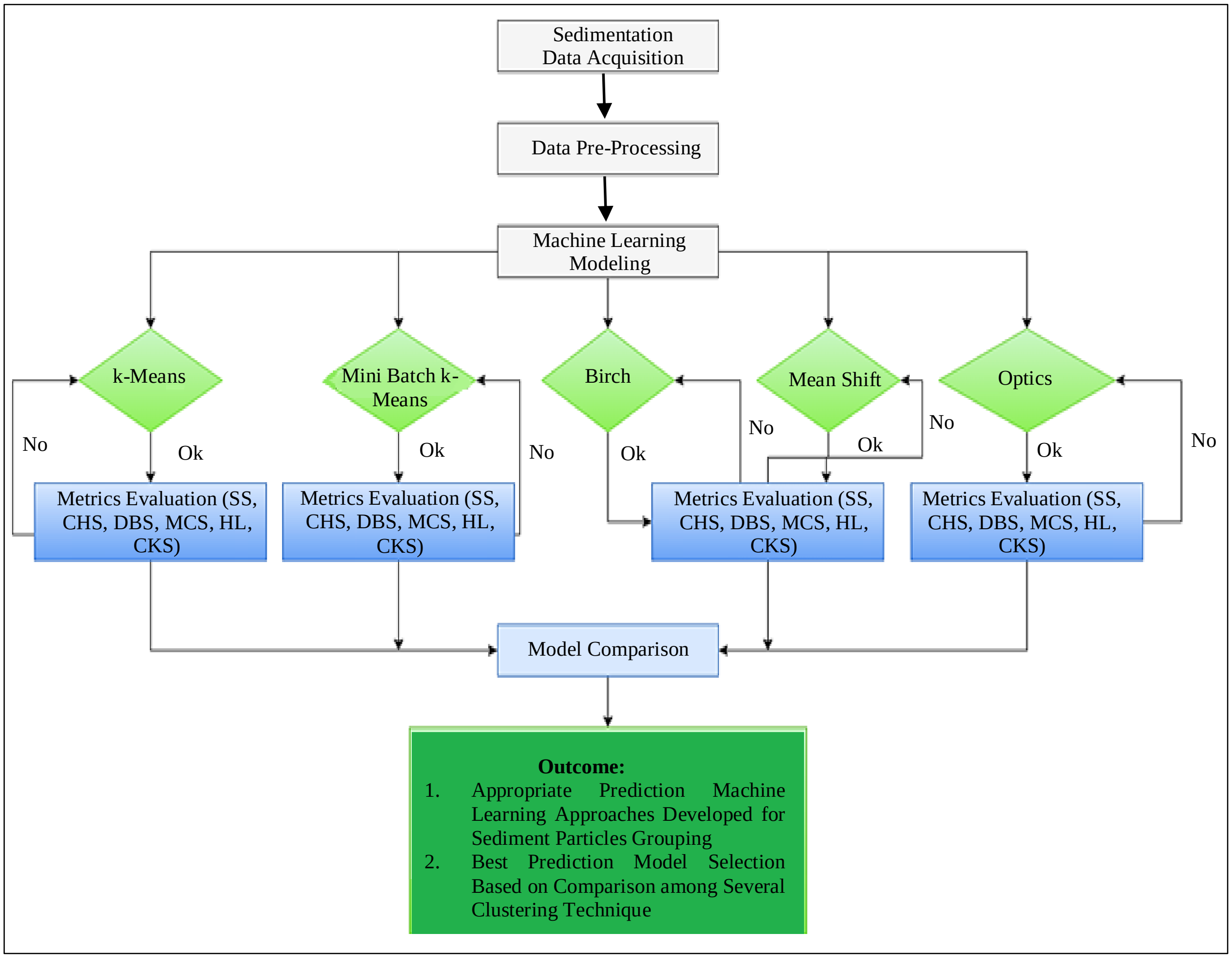

Sedimentation Data Acquisition
Data Pre-Processing
Machine Learning Modeling
k-Means
Mini Batch k-Means
Birch
Mean Shift
Optics
No
Ok
Metrics Evaluation (SS, CHS, DBS, MCS, HL, CKS)
Model Comparison
Outcome:
1. Appropriate Prediction Machine Learning Approaches Developed for Sediment Particles Grouping
2. Best Prediction Model Selection Based on Comparison among Several Clustering Technique

Fig. 3 Workflow for sedimentation grouping

3.4. Reduction of Duplicate Rows

Here, the replicated rows are removed so that the machine learning algorithms might be impossible to mug the data relevant to the sedimentation procedure. However, the duplicated rows are not present; hence, similar rows and columns are generated for performing the next phase.

3.5. Eradicate Skewness, Infinity, and NaN

3.5.1. Skewness

Skewness is a metric for the asymmetrical of a real-valued random variable's stochastic process with respect to its mean. Impartial skew over the specified axis is returned by the skew () function using N-1 normalization. A statistical model provides the most prevalent sort of information and probabilistic distribution. The formula described here is to find the skewness.

$$Skewness = \frac{3\,(Mean - Median)}{Standard\ Deviation} \tag{1}$$

The reason behind the skewness removal is that the data should be transformed to follow a normally distributed curve by removing outliers because a simple regression analysis was fitted based on a statistical distribution. The relative probability of the parameters is unrelated to the constraints when examining predictor variables.

3.5.2. Infinity

Numerous individuals believe that ∞ is simply a very massive proportion. However, in fact, infinity refers to the notion that something is limitless and eternal.

Additionally, Numpy provides the framework for indefinite values of its own. Moreover, arrays containing unbounded numbers are a possibility. Hence, the infinity values are removed.

3.5.3. NaN-Not A Number

Here, the authors replace NaN values in a DataFrame with a completely empty or Vacant word by employing the replace () or fillnan () functions. Not A Number, or NaN, represents one of the standard ways to indicate the amount of incomplete information in a Numpy DataFrame. In Python, the function df. Replace (np.nan, 0) is used to replace the NAN values.

3.6. Build Machine Learning Models

After the elimination of skewness, infinity, and Nan, the number of data comprises 1,11,045 rows with the quantity of features 7, then fed into a machine learning-based clustering technique for training the data, which predicts the outcome in the grouping of sedimentation particles. The workflow of this research work is depicted in Figure 3. The phases of this research work are explained in the step-by-step procedure in Section 3.

Initially, the sedimentation dataset is collected, and then pre-process the data is applied machine learning clustering approach for sediment particles such as gravel, sand, mud, and gravel; then, based on score metrics, the model performance is evaluated in grouping sedimentation particles automatically. Clustering is the division of massive amounts of data into more manageable chunks. It is a challenge in unsupervised learning.

To obtain data from a relevant feature or domain, such as comparable usage patterns in a client list, grouping is typically used in an investigation. Here, the authors discussed the unsupervised machine-based clustering approaches, which explains how these algorithms are appropriate for grouping sediment particles based on auto-grouping.

3.7. K-Means Clustering

This method is a simple and common unsupervised ML approach. Unsupervised methods generally draw conclusions from databases utilizing just model parameters not considering predetermined or clearly labelled results. Finding correlations by combining comparable data points is the straightforward goal of K-means. K-means seeks a database for such the predefined set (k) of clusters to obtain this objective.

A cluster is a data group that is integrated because of its shared features. The position that, whether actual or virtual, serves as the centroid of clusters. Every information point was allotted to a specific group by minimizing the overall square among every group. Data mining's K-means method utilizes an initial set of randomly picked clustering as the starting points for each cluster to analyze the training inputs. From there, repeated (repeatable) computations are made to optimize the locations of the cluster centers. Whenever either one of the following presents:

- The cluster centers have stabilized their values remain unvaried because the clusters are effective.
- Iterations have reached the predetermined value.

Silhouettes factor in k-means: The silhouettes factor has a range among [-1, 1]. When the parameter receives a score of 1, it means that its cluster is the largest and that it exists furthest apart from all other clusters. The very worst number is one. Scalability to 0 indicates groupings that intersect. The objective function of the k-means is given in the following equation,

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(x_{ij} - c_i\right)^2 \tag{2}$$

Here, J was the objective function to be minimized, k was the total clusters, n_i was the total data points in cluster i, x_{ij} was the jth data point in cluster i, and c_i was the centroid of cluster i.

$$argmin_i\left(x_j - c_i\right)^2 \tag{3}$$

Using this equation, assign all the points x_j to the cluster with the nearest centroid.

$$c_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij} \tag{4}$$

Update the centroid of all the clusters as the mean of its allotted data points.

3.8. Mini-Batch K-Means

The primary concept behind the Mini Batch K-means technique is to employ short, rectified randomized groups of input to enable them to be retained in memory. A fresh representative selection from the database is taken during each repetition and utilized to construct the clustering. This process is continued until it converges.

The benefit of this technique is that it uses a resolved random sample instead of the entire database throughout every repetition, which lowers the computing cost. The coefficient of Silhouette evaluates how the sediment particles are closer to its own cluster in comparison with the remaining clusters which means perhaps an additional cluster that can equally or more accurately describe it.

3.9. BIRCH Clustering

A different classification technique than Mini batch K-Means is offered: BIRCH is abbreviated as Balanced Iterative Reducing and Clustering Hierarchies. The numbers of clusters are known from the branch as the information was converted to tree-like structures. Such a number of clusters can also serve as the source for many other group techniques like hierarchical clustering or perhaps the optimal group centroids.

The grouping BIRCH splits the database into smaller summaries initially, and hence groups the short highlights. The database was not quickly clustered by it. Due to the summary could then be clustered by more clustering approaches after being created, BIRCH was frequently utilized in integration with some more clustering methods.

BIRCH was a scaling cluster algorithm on the basis of hierarchical grouping that is quick when operating with massive data since it only needs to read the database once. The CF (clustering features) tree was the backbone of this model. Moreover, this model builds groups utilizing the tree-structured summation.

This model generates a tree structure for the provided data, which it refers to as a Clustering Feature (CF) tree. The technique compressed the input into pairs of CF links in relation to the CF structure. CF sub-clusters are nodes that contain numerous sub-clusters. Such groups are without terminals in cluster feature nodes. The algorithm has four stages, namely:

- Reading information into memory.
- Data condensing (resize data).
- Worldwide clustering
- Cluster refinement.

Resizing input and optimizing groups are two of these four processes that are accessible. When greater clarification is needed, they enter the process. In contrast, putting data into a model is like reading the inputs. Following data loading, the algorithm reads all the data and fits it into CF trees. When using traditional clustering methods, it transmits CF trees for large-scale clustering. The issue with CF trees in which the identical valued points are assigned to various leaf nodes is finally resolved by refining.

The features used in this approach are threshold, in which the highest quantity of data samples is assigned in subgroups of the CF tree, and branching factor defines the factor which denotes the sub groups and number of clusters. The CF can be computed using $CF = (N, LS, SS)$, where N was the total data points in the cluster, LS was the linear sum of data points and SS was the squared sum of data points. The CF tree structures the data into nodes, with all nodes representing a cluster feature. In the CF tree traversal, BIRCH generates subclusters based on specific criteria, creating a hierarchical structure.

3.10. OPTICS Clustering

OPTICS is abbreviated as Ordering Points To Identify Clustering Structure, the density-based clustering approach to point out the clusters in various locations among input space. Because it utilizes a sorted list (Minimum Memory) to choose the following data item based on Routing Proximity that is nearest to the node currently under-examined, the OPTICS clustering approach uses greater space.

The epsilon argument is not necessary for the OPTICS clustering algorithm, which is just used in the pseudo-code to shorten the computation time. The analysis measures of the method are more accurate and are subsequently made simpler. OPTICS does not cluster the input data. The coder must examine the reachability range plot it produces and group the spots appropriately.

3.11. Hybrid Clustering

In this research, the proposed hybrid clustering approach strategically combines the strengths of K-means clustering and BIRCH clustering to optimize the accuracy and efficiency of grouping underwater sediment particles. The process initializes with the initialization of cluster centroids for K-means clustering. These centroids can be chosen randomly or based on specific criteria. Simultaneously, the BIRCH clustering method is initiated, constructing a CF tree for the input data.

The K-means clustering process begins with the assigned centroids, where data points are grouped into clusters by minimizing the total sum of squared distances within each cluster. Centroids are iteratively updated until convergence, signifying stability or reaching a predetermined number of iterations. The CF tree structure generated by BIRCH clustering is seamlessly integrated into the K-means clustering process. This involves exploiting the information stored in the CF tree to refine the clustering derived from K-means. The CF tree's hierarchical structure aids in capturing complex relationships and patterns in sediment particles.

The hierarchical nature of BIRCH clustering provides additional insights into the data structure. The CF tree organizes data into clusters and sub-clusters, offering a detailed view of relationships between sediment particles. This hierarchical information enhances the accuracy of clustering, especially when dealing with nested or hierarchical groupings.

The parameter tuning stage in BIRCH clustering is crucial for addressing issues related to identical-valued points assigned to different leaf nodes. This step fine-tunes the clustering results from K-means, aiming for improved accuracy by eliminating ambiguities or inconsistencies. The hybrid approach involves tuning parameters such as the number of clusters, threshold for data samples in subgroups of the CF tree, and branching factor.

Parameter tuning is crucial for optimizing the performance of the hybrid clustering approach, ensuring adaptability to the unique characteristics of underwater sediment particle data. Overall, the hybrid clustering method integrates the iterative refinement of K-means clustering with the hierarchical structure and summarization capabilities of BIRCH clustering. This integration is designed to provide a robust and accurate solution for automatically grouping underwater sediment particles in the sea, as discussed in this research work. The initialization of the k-means is updated with the following equation,

$$J_{k-means} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(x_{ij} - c_i\right)^2 \quad (5)$$

$$J_{Birch} = \sum_{i=1}^{n} \sum_{j=1}^{m_i} dist\left(x_{ij}, c_i\right) \quad (6)$$

Here, $J_{k-means}$ was the k-means objective function, J_{birch} was the BIRCH objective function, and $dist(x_{ij}, c_i)$ was the distance among the data points x_{ij} and cluster centroid c_i.

Mapping the centroids obtained from the BIRCH clustering to the nearest centroids obtained from k-means using $argmin_i dist(c_{birch}, c_{kmeans})$. Allot the data points to clusters according to the mapped centroids, which presents the final output.

```
Initialize Function to Implement Hybrid Clustering
define hybrid_clustering(data, k_kmeans, threshold_birch):
Apply K-means clustering
    kmeans = KMeans(n_clusters=k_kmeans)
    kmeans_labels = kmeans.fit_predict(data)
Identify cluster centroids from K-means
    cluster_centroids = kmeans.cluster_centers_
Apply Birch clustering to refine clusters
    birch=Birch(threshold=threshold_birch, n_clusters=None)
    birch_labels = birch.fit_predict(data)
Map Birch labels to the nearest K-means cluster
    birch_cluster_centroids = np.array([data[birch_labels == i].mean(axis=0) for i in range(birch.n_clusters_)])
kmeans_labels_mapped,=pairwise_distances_argmin_min(birch_cluster_centroids, cluster_centroids)
Assign final labels based on Hybrid Clustering
    final_labels = np.zeros_like(kmeans_labels)
    for i in range(len(kmeans_labels)):
   final_labels[i]=kmeans_labels_mapped[birch_labels[i]]
    return final_labels
```

4. Results and Discussion

This section discussed the evaluation outcomes of sedimentation particle grouping based on unsupervised clustering approaches. The experimental analysis of the proposed clustering methodologies is conducted with an implementation in Python, using Google Colab Pro as the computing platform. The experiments involve the application of various clustering and the proposed hybrid clustering approach to underwater sediment particle datasets. Python's scikit-learn library is instrumental in implementing these clustering techniques, ensuring a complete and standardized methodology for comparison.

Throughout the experimental analysis, various performance metrics, including Silhouette Score, Calinski-Harabasz, Davies-Bouldin, Matthew's Correlation Score, Hamming Loss, and Cohen Kappa score, are systematically employed. These metrics serve as quantitative indicators to evaluate the efficiency of the clustering algorithms in automatically grouping underwater sediment particles.

4.1. Performance Evaluation

Table 2 discusses the score metrics, including Silhouette (SS), Davies Bouldin (DB), Mathew's Correlation (MC), Calinski Harabasz (CH), Hamming Loss (HL), and Cohen Kappa (CK) scores are evaluated for all clustering techniques to predict the overall performance of clustering techniques. The K-means clustering automatically groups the range of sedimentation data for each column, and finally, the output in the visualization form is depicted in Figure 4.

Silhouette Score is used to identify whether the clustering method is correctly grouped or not. In visualization, if more line goes to below 0 that algorithm is poor. Figure describes the silhouette score for k-means with 111045 samples with seven features for clustering the data to group the sediment particles such as gravel, rock, sand, surface carbon and nitrogen.

Figure 5 depicts the silhouette score of k-means appropriate for checking whether the clustering approach performs correct clustering or not. The mini-batch clustering algorithm is appropriate for grouping the input data randomly with a constant size that might be stored in memory.

A new has been chosen from the database randomly to update the clusters, which has been repetitive till converged gent, as depicted in Figure 6. Based on the silhouette score values, the performance of the mini-batch k-means clustering method is found. The plot of such score is depicted in Figure 7, in which the x-axis represents the relevant coefficient values, and the y-axis belongs to cluster labels for identifying the average silhouette score.

The importance of th**e** BIRCH clustering method is to group huge databases by producing small and compressed summaries of such huge databases stayed as data. Figure 8 depicts the grouping of sedimentation particles such as mud, sand, gravel, rock, surface carbon and nitrogen using this BIRCH clustering. The authors formed 7 clusters labelled in Figure 9 using BIRCH to estimate the silhouette score to predict the performance of the clustering technique.

Table 2. Performance evaluation based on several scores

Model	SS	CH	DB	MC	HL	CK
Hybrid Clustering	0.5426	49694.11	0.5892	0.6500	0.0125	0.8038
K-Means	0.5278	48293.55	0.6030	0.3493	0.9995	-0.0581
Mini Batch k-Means	0.5151	46512.009	0.6220	0.1953	0.6215	0.1862
BIRCH	0.4919	44681.28	0.6336	-0.1419	0.9724	-0.0541
OPTICS	-0.6575	86.102	1.6298	0.1199	0.3549	0.1159

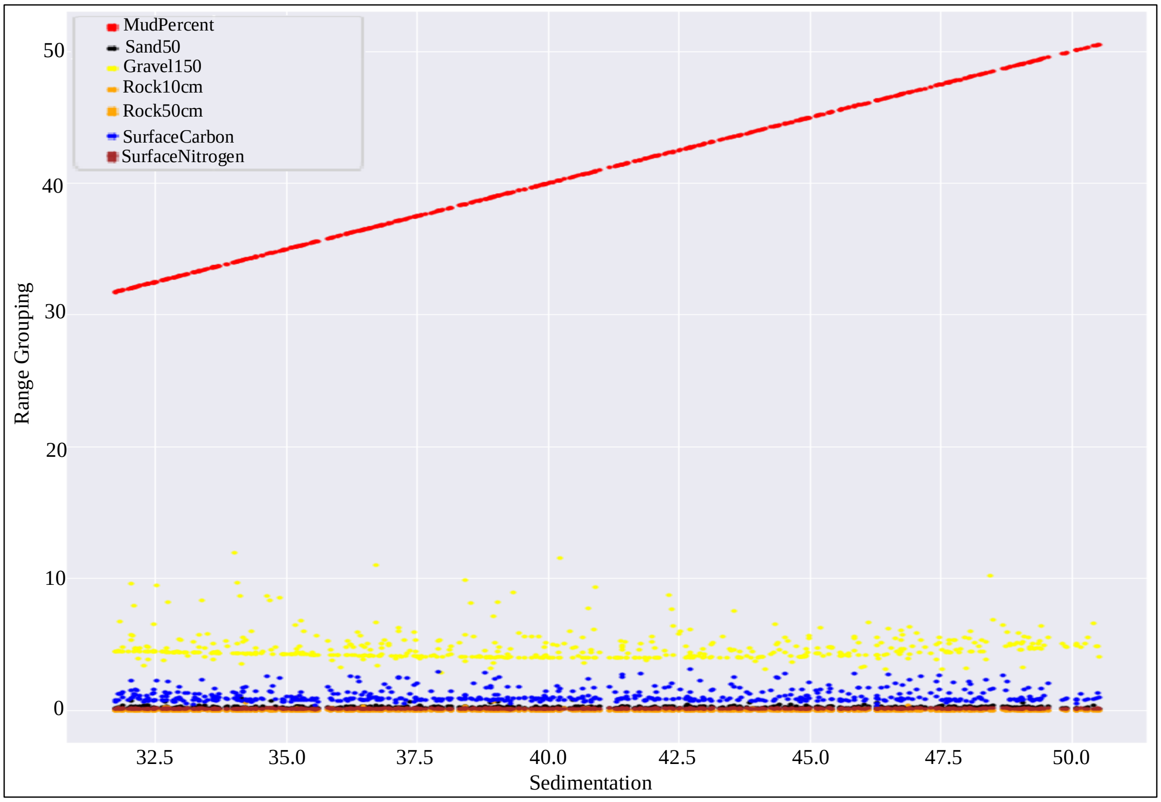


Fig. 4 Clustering sediment particles using k-means

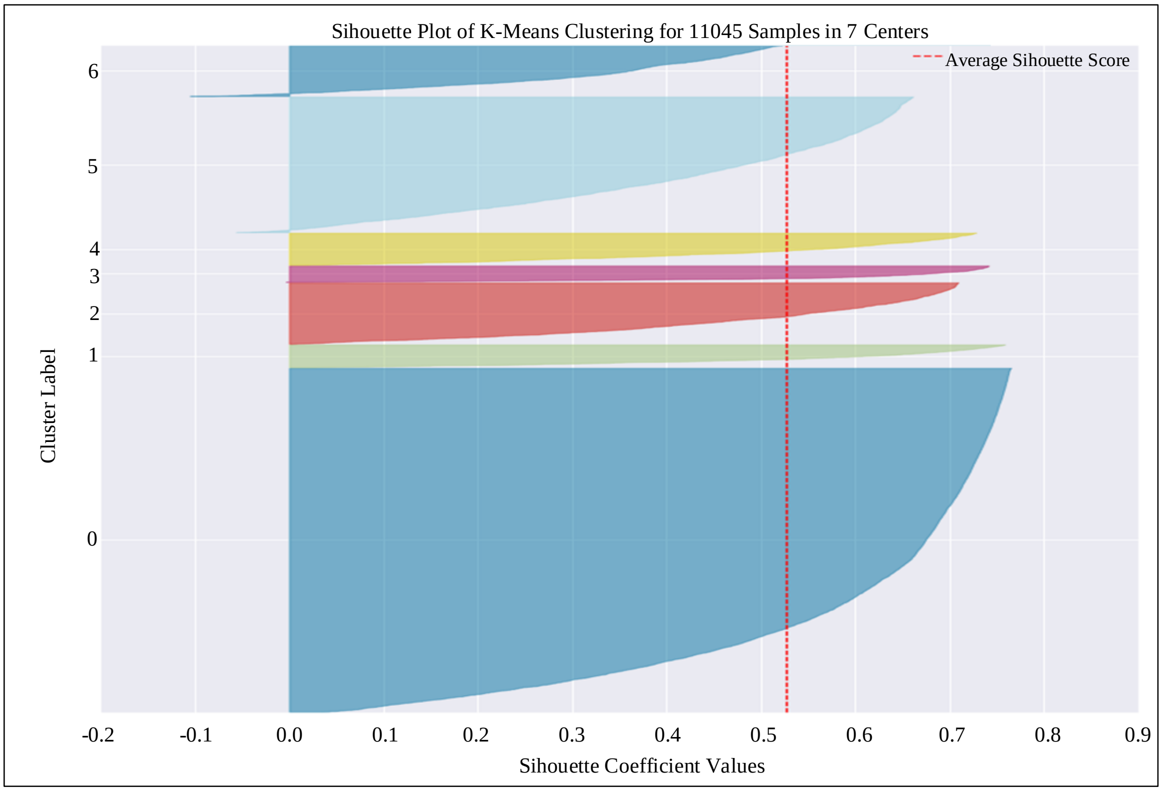


Fig. 5 Silhouette score graph using k-means

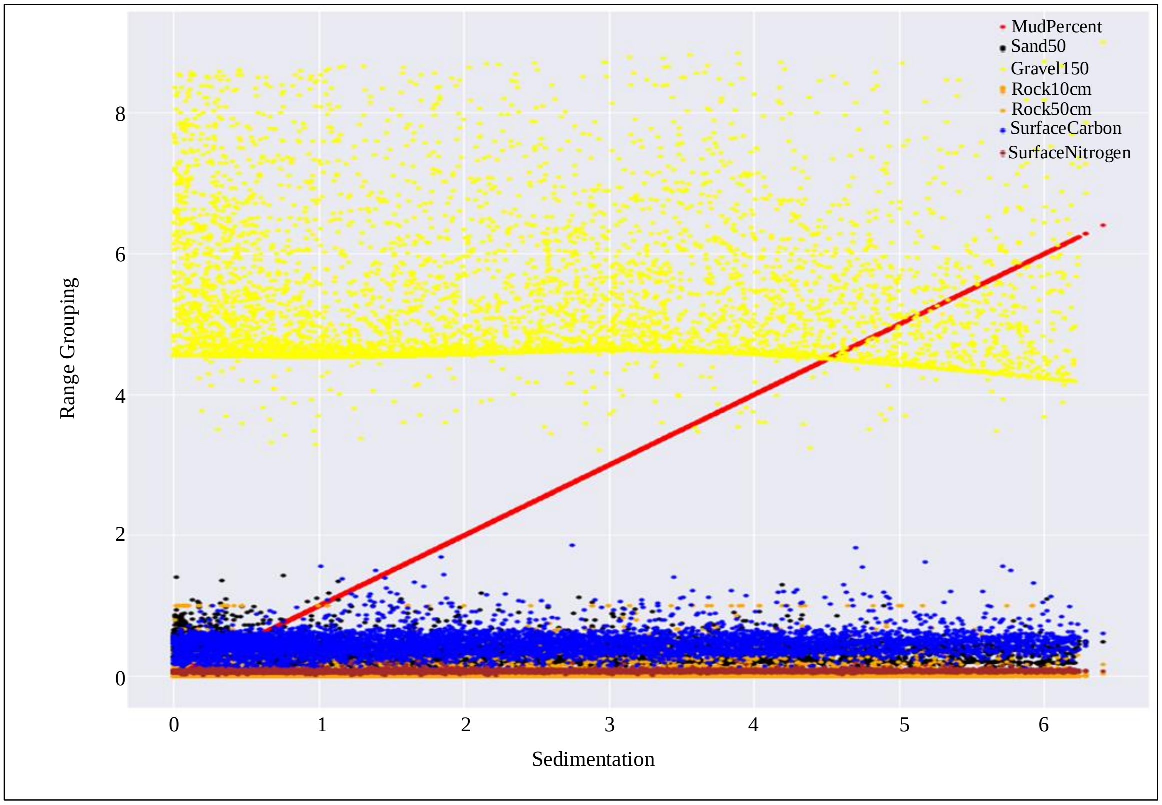


Fig. 6 Grouping sedimentation using mini batch clustering

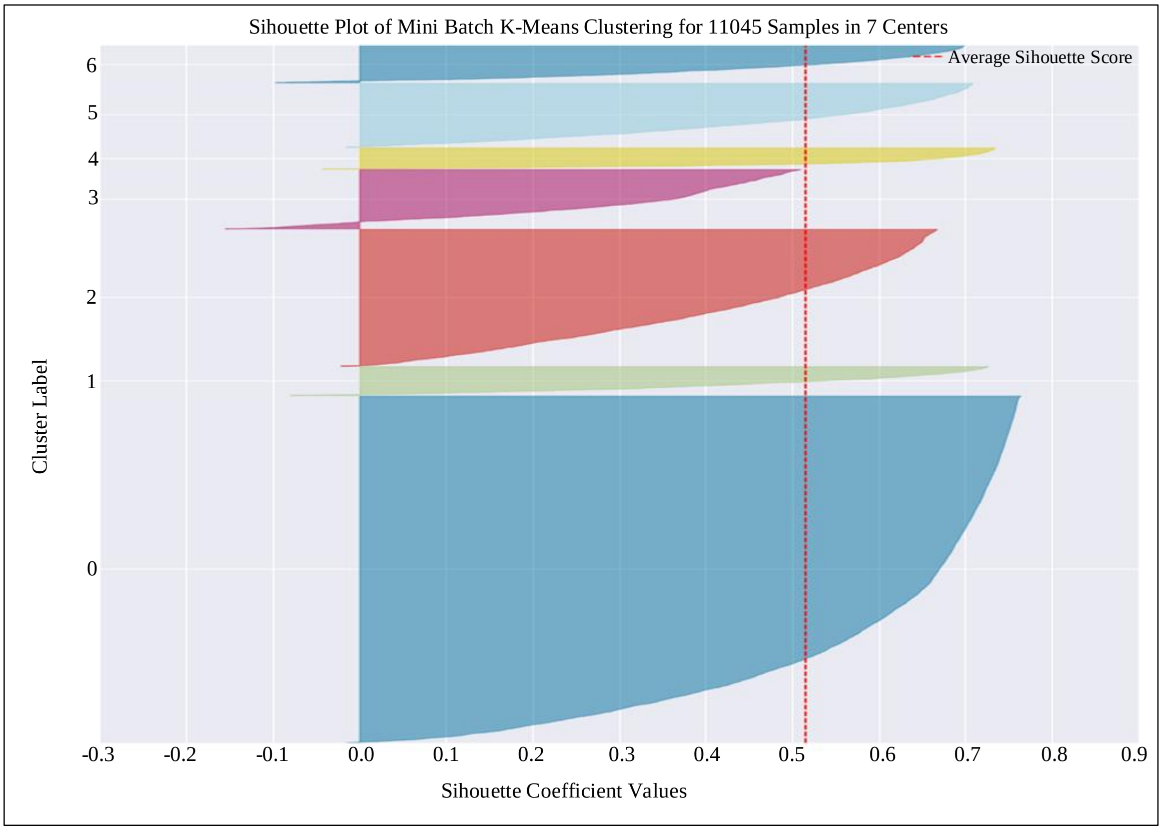


Fig. 7 Graph for Silhouette score of mini batch method

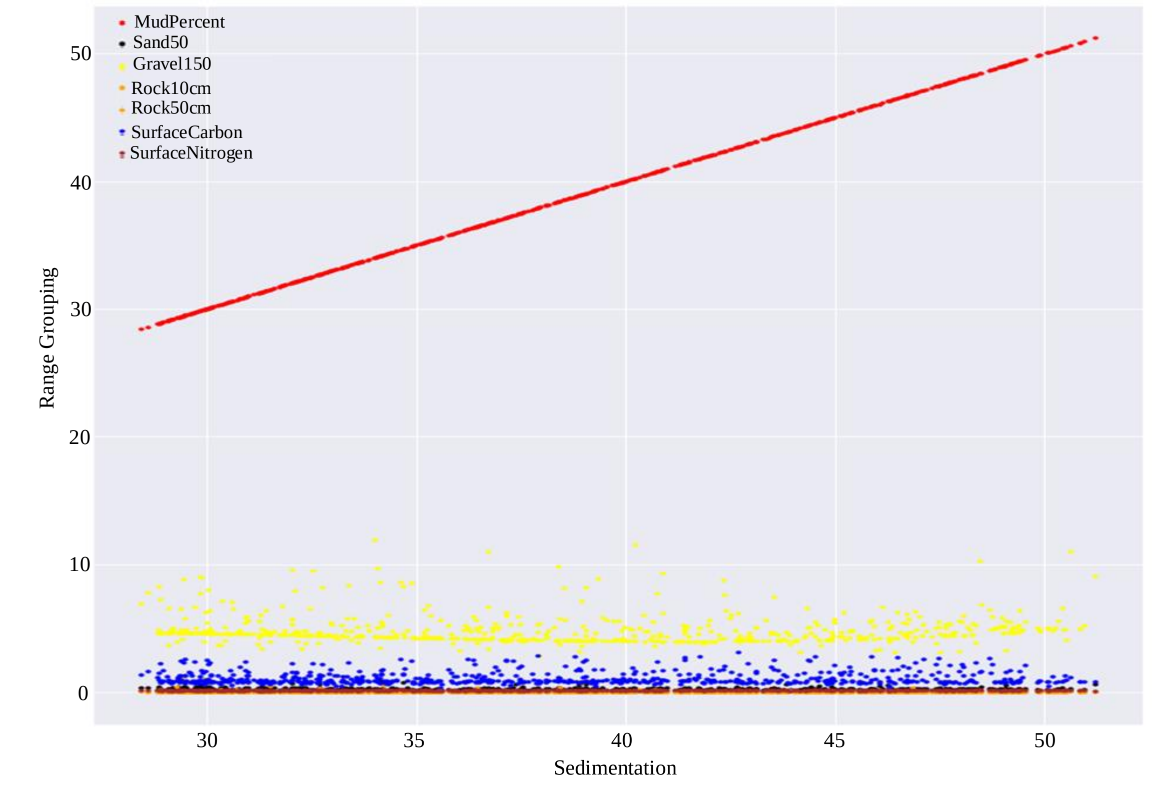


Fig. 8 BIRCH clustering method for sediment grouping

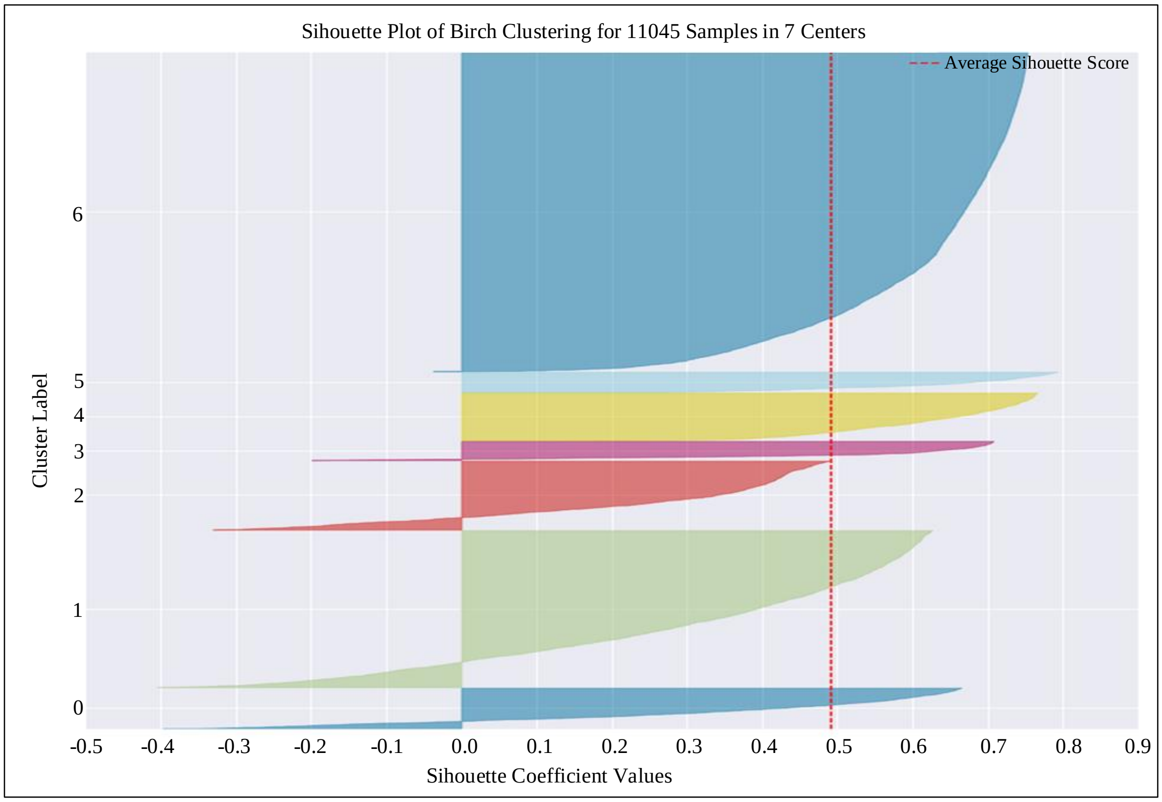


Fig. 9 Plot for Silhouette score using BIRCH clustering

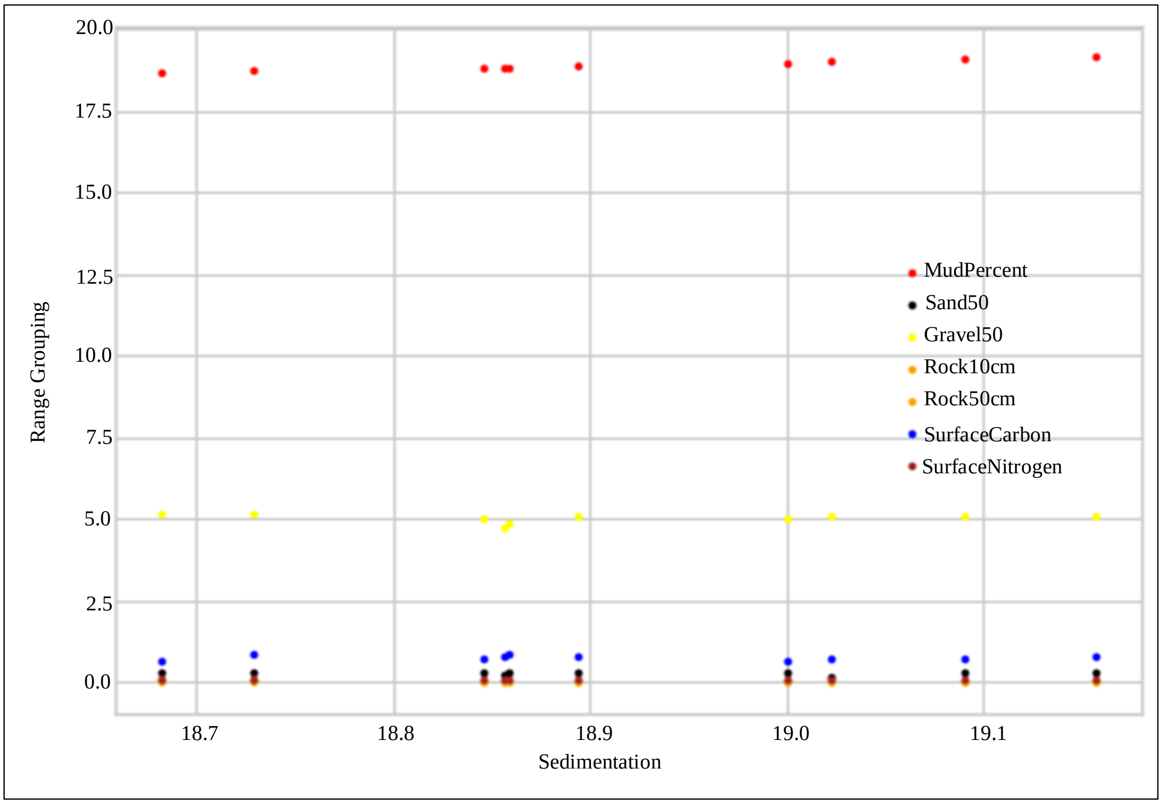


Fig. 10 Sedimentation particles grouping plot using OPTICS clustering

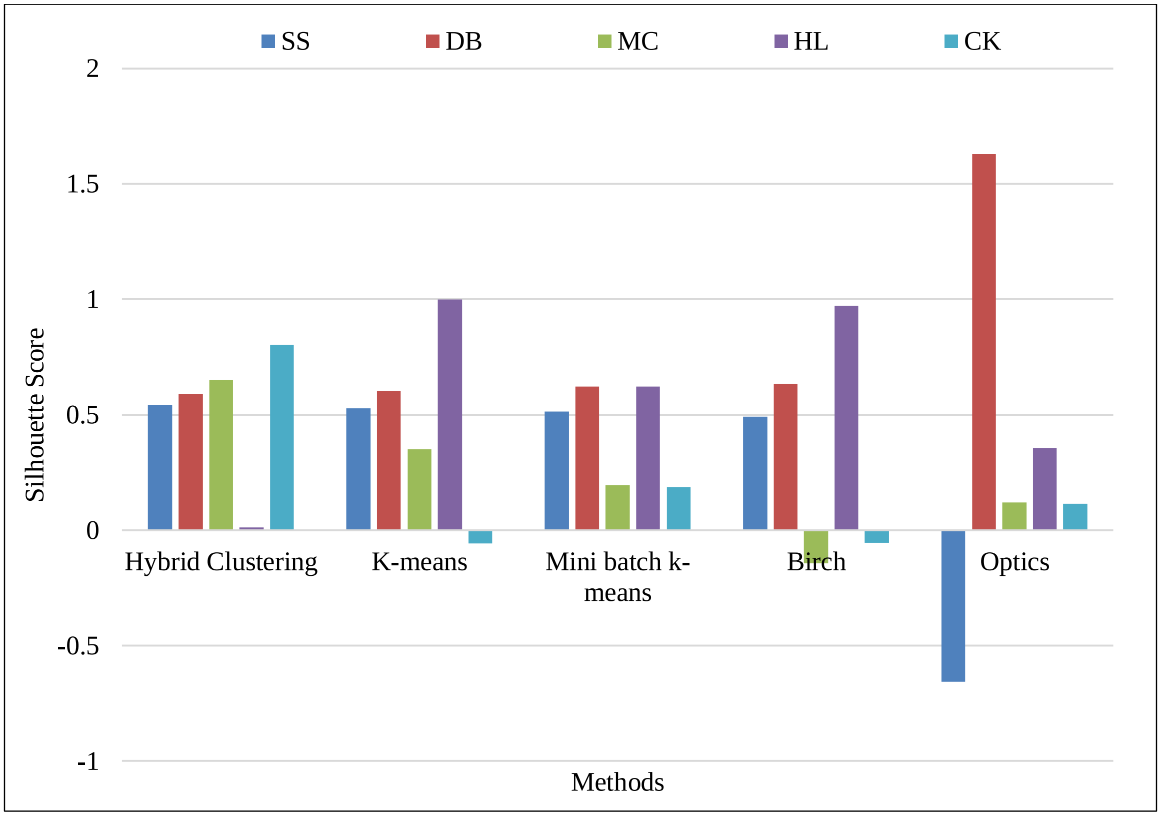


Fig. 11 Comparison of SS of the hybrid clustering model

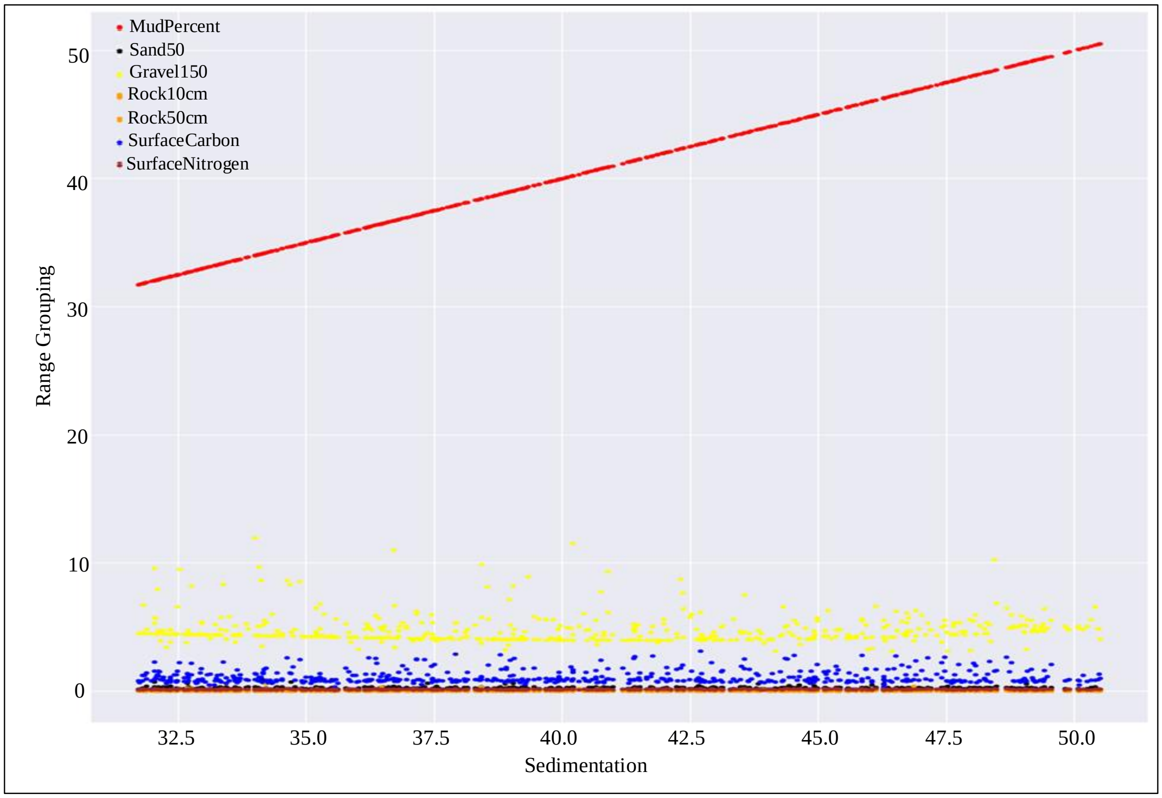


Fig. 12 Hybrid clustering method for sediment grouping

In hybrid clustering correlation coefficient gives a positive 0.65 score. Compared to all clustering models, this score is higher; finally, using the correlation coefficient score, the hybrid clustering performs well. In the case of the Cohen kappa score, the score is lesser; hence, the performance is good. If the score is higher, then the performance is better in the case of hamming loss metrics; the OPTICS algorithm performs well in these metrics. Hybrid clustering is performed well but OPTICS performs some greater perform well.

Finally, it is found that hybrid clustering is a good algorithm for this dataset compared to all algorithms of metrics. The average silhouette score is predicted for each formed cluster mentioned as red dotted lines. The grouping among sedimentation particles such as MudPercent, sand, gravel, rock, surface carbon, and surface nitrogen has been implemented using the OPTICS clustering method is depicted in Figure 10.

The grouping among sedimentation particles such as MudPercent, SandD50, GravelD50, Rock10cm, Rock50cm, SurfaceCarbon, and SurfaceNitrogen have been implemented using the hybrid clustering method, which is depicted in Figure 12. By comparing all clustering methods, hybrid clustering gives the best performance.

While comparing to all clustering approaches, silhouette visualization shows best in hybrid clustering. The score of the silhouette and Calinski Harabasz and Davies Bouldin in hybrid clustering was high score compared to all scores. If the correlation coefficient score attains a positive score, then that algorithm is considered to be good.

The hybrid clustering outperforms all the other models in terms of silhouette score. The hybrid clustering approach demonstrates a commendable silhouette score, indicating well-defined and distinct clusters compared to several other models.

The hybrid clustering model exhibits significantly higher cluster quality, as indicated by the Calinski Harabasz score, highlighting its efficacy in capturing meaningful patterns in the data. The hybrid clustering approach excels in cluster compactness and separation, offering better-defined clusters than select competing models, as verified by Davies Bouldin's score.

The hybrid clustering model shows a higher Mathew's Correlation score, indicating its robustness in capturing true positives and minimizing false positives and false negatives. The hybrid clustering approach minimizes misclassifications and exhibits the lowest Hamming Loss, showcasing its accuracy in predicting sediment particle groupings. The hybrid clustering model attains the highest Cohen Kappa score, indicating strong agreement between predicted and actual cluster assignments.

5. Conclusion

The research presented an analysis and comparison among some clustering algorithms most used on sedimentation grouping automatically. The authors search for groupings that could be simply characterized by the cluster centers because the grouping can assist in simplifying the intricate architecture of sediment particles.

Hence, several clustering approaches like k-means, mini batch k-means, BIRCH, and OPTICS clustering are implemented for grouping the sediment-based particles. Moreover, SS, CH, DB, MC, HL, and CK scores with n=7 are evaluated to predict the performance of the unsupervised clustering method.

From the analyzed clustering models, this research introduces a hybrid clustering model for automated underwater sedimentation analysis, combining the strengths of K-means and BIRCH Clustering. Evaluating performance metrics, the hybrid model consistently outperforms traditional clustering methods. With a higher Mathew's Correlation Score, lower Hamming Loss, and superior Cohen Kappa Score, the hybrid clustering approach proves its robustness in accurately grouping sediment particles. The hybrid clustering model has the best results with 0.5426 SS, 49694.11 CH, 0.5892 DB, 0.6500 MC, 0.0125 HL, and 0.8038 CK. Among those implemented clustering approaches, the hybrid method generates better outcomes in auto-grouping the sediment particles, such as gravel, rock, etc., in underwater acoustics. This research contributes an efficient model for precise environmental monitoring, offering scalability and efficiency in underwater ecosystem studies. Future work may involve parameter optimization and real-world validation to enhance practical utility. The hybrid clustering model emerges as a significant advancement, promising transformative impacts on automated sedimentation analysis.

Acknowledgments

The authors would like to thank the Vels Institute of Science, Technology, and Advanced Studies, Chennai, Tamilnadu, India, for their support and motivation throughout this research.

References

[1] W.K. Stewart, M. Jiang, and M. Marra, "A Neural Networks Approach to Classifications of Side Scans Sonar Images from Midocean Ridges Areas," *IEEE Journals of Oceanic Engineering*, vol. 19, no. 2, pp. 214-224, 1994. [CrossRef] [Google Scholar] [Publisher Link]

[2] Xiaowen Luo et al., "Sediments Classifications of Small-Sizes Sea Bed Acoustics Image Using Convolution Neural Network," *IEEE Access*, vol. 7, pp. 98331-98339, 2019. [CrossRef] [Publisher Link]

[3] Zaki Abda et al., "Suspended Sediments Loads Simulations during Floods Event Using Intelligence System: Case Study on the Semiarid Region of Mediterranean Basins," *Water*, vol. 13, no. 24, pp. 1-19, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Dillip K. Ghose, and Sandeep Samantaray, "Sedimentations Process and Its Assessments through Integrated Sensors Network and Machine Learning Process," *Computational Intelligence in Sensor Networks*, pp. 473-488, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] Maryam Asadi et al., "Predictions of Rivers Suspended Sediments Loads Using Machines Learning Model and Geo-Morphometric Parameter," *Arabian Journals of Geoscience*, vol. 14, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Vahid Nourani, Huseyin Gokcekus, and Gebre Gelete, "Estimations of Suspended Sediments Loads Using Artificials Intelligences-Based Ensembled Models," *Complexity*, vol. 2021, pp. 1-19, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Mohammad Zounemat-Kermani et al., "On the Complexity of Sediments Loads Modelling Using Integrative Machines Learning: An Applications to the Great Rivers of LoÃza in Puerto-Rico," *Journals of Hydrology*, vol. 585, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Banu Yilmaz et al., "Estimating Suspended Sediments Loads with Multi Variate Adaptive Regressions Splines, Teaching-Learning Based Optimizations, and the Artificial Bee Colony Model," *Sciences of the Total Environments*, vol. 639, pp. 826-840, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[9] Nouar AlDahoul et al., "Suspended Sediments Loads Predictions Using Long Shorts-Terms Memory Neural Networks," *Scientifics Report*, vol. 11, pp. 1-12, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Tim Berthold et al., "Seabed Sediments Classifications of Sides-Scans Sonar Data Using Convolution Neural Network," *2017 IEEE Symposiums Series on Computational Intelligences (SSCI)*, Honolulu, USA, pp. 1-8, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] Cheng-Chia Huang et al., "Real-Time Forecasting of Suspended Sediment Concentrations in Reservoirs by the Optimal Integration of Multiple Machine Learning Techniques," *Journals of Hydrology: Regionals Studies*, vol. 34, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Aditi Awasthi, S.S. Lokhande, and M. Selva Balan, "Acoustics Signals Processing Techniques to Classify Reservoirs Sediment," *International Research Journal of Engineering and Technology*, vol. 3, no. 6, pp. 2817-2819, 2016. [Publisher Link]

[13] Xiaodong Cui et al., "Deep Learning Models for Seabed Sediments Classifications Based on Fuzzy Ranking Features Optimizations," *Marine Geology*, vol. 432, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Issam Mohamed, and Imtiaz Shah, "Suspended Sediments Concentrations Modelling Using Conventional and Machines Learning Approach in Thames Rivers, London," *Journals of Water Managements Modelling*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] B. Bhattacharya, and D.P. Solomatine, "Machine Learning in Sedimentations Modeling," *Neural Network*, vol. 19, no. 2, pp. 208-214, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[16] Haoran Liu et al., "Sediments Identifications Using Machine Learning Classifier in a Mixed-Textures Dredge Pits of Louisiana Shelfs for Coastal Restorations," *Water*, vol. 11, no. 6, pp. 1-18, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Senlin Zhu et al., "Machine Learning Approaches for Estimations of Sediments Settling Velocity," *Journal of Hydrology*, vol. 586, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Maheswar Ojha, and Saumen Maiti, "Sediments Classifications Using Neural Network: An Example from the Sites-U1344A of IODP Expeditions 323 in the Bering Sea," *Deep Sea Research Part II: Topical Studies in Oceanography*, vol. 125-126, pp. 202-213, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[19] P.J. Mitchell et al., "Sedimentations Rate in the Baltic Sea: A Machine Learning Approach," *Continental Shelfs Research*, vol. 214, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Ashish Mishra, and Santonu Goswami, "*Analyzing Seasonal and Inter-Annual Turbidity of a Wetland Ecosystem in India Using Machine Learning and Time-Series Modeling*," Indian Space Research Organization, KIET Institutions, Research Report, 2022. [Google Scholar] [Publisher Link]

[21] Xiaoming Qin et al., "Optimizing the Sediments Classifications of Small Side-Scans Sonar Image Based on Deep Learning," *IEEE Access*, vol. 9, pp. 29416-29428, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[22] G. Revathy, P. Sathish Kumar, and Velayutham Rajendran, "Development of IDS Using Mining and Machine Learning Technique to Estimate DoS Malwares," *International Journal of Computational Sciences and Engineering*, vol. 24, no. 3, pp. 259-275, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Viviana Siless et al., "A Comparison of Metric and Algorithm for Fiber Clustering," *2013 International Workshops on Patterns Recognitions in Neuroimaging*, Philadelphia, USA, pp. 190-193, 2013. [CrossRef] [Google Scholar] [Publisher Link]