Using Clustering Techniques to Understand Student Involvement in an Online Environment

Vandana Naik[1,2,*] and Venkatesh Kamat[3]

[1]Goa Business School, Goa University, Goa, India
[2]Centre for Research, Development and Innovation, Goa State Higher Education Council, Directorate of Higher Education, Goa, India
[3]School of Mathematics and Computer Science, Indian Institute of Technology, Goa, India
Email: dcst.vandana@unigoa.ac.in (V.N); vvkamat@iitgoa.ac.in (V.K.)
*Corresponding author
Manuscript received August 13, 2024; revised September 4, 2024; accepted January 3, 2025; published May 21, 2025

***Abstract*—Student engagement in online learning environments is critical in improving educational outcomes and instructional strategies. Previous studies on engagement patterns using online log datasets often focus on interaction frequency, neglecting intensity and comprehensive activity coverage. This study addresses these gaps by introducing a novel approach grounded in the Community of Inquiry (CoI) model to calculate engagement parameters. The research objectives include deriving meaningful engagement metrics, clustering students based on these metrics, and evaluating clustering algorithms to identify the most effective method. The methodology involves processing Moodle log data to extract three key engagement parameters: Number of sessions, session duration, and engagement levels encompassing social and cognitive dimensions. These derived parameter values were then compared to the labels set manually by two raters. High agreement (0.9409 correlation) between these two methods validates the algorithm's efficiency and reliability in measuring student engagement. Next, clustering algorithms, such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Gaussian Mixture Model (GMM), K-means, Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), etc., are applied to group students, with cluster quality assessed using indices like Davies-Bouldin, silhouette coefficient, and Calinski-Harabasz. The findings reveal that K-means and Birch algorithms effectively categorize students, with the CoI-derived engagement parameters proving to be the most influential. These insights highlight the critical role of cognitive and social interactions in engagement and demonstrate the superiority of such methods in discovering patterns in student data. This study provides a robust framework for analyzing student engagement, offering actionable insights for educators to enhance online learning experiences.**

***Keywords*—engagement, agglomerative hierarchy clustering algorithm, K-means, Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Gaussian Mixture Model (GMM)**

I. Introduction

In recent years, online learning has gained significant momentum, with many educational institutions adopting Learning Management Systems (LMS) to facilitate remote education. Moodle, one of the most popular LMS platforms, generates vast data by capturing student interactions with the system. Analysing this data holds immense potential for understanding student engagement patterns and, in turn, improving student's learning experience. Student engagement is a multifaceted concept that encompasses various dimensions, including participation, attention, interaction, and motivation. It plays a vital role in academic success, knowledge retention, and overall learning outcomes. Therefore, gaining insights into student engagement behaviours is crucial for educators and administrators to enhance teaching methodologies, personalise instruction, and design interventions to support student learning [1]. Many studies have been conducted over the past few decades that have utilized various methods like self-report questionnaires, teacher rating/field observations, computer vision-based methods, Physiological sensors, log analysis, etc. These methods suffer from drawbacks like bias, false reporting, hardware dependency and scalability issues. The category of log analysis does not have many of these problems. Still, the research work conducted in this category is limited with respect to the parameters used in calculating the levels. This research proposes a technique that relies only upon the logs generated in the online learning environment to derive engagement-related parameters. It explores the effectiveness of this method by comparing it with values obtained from manual labelling of the same logs.

Clustering techniques offer a powerful analytical approach to uncovering meaningful patterns and segments within student datasets from Moodle logs. By applying clustering algorithms, researchers can identify distinct groups of students based on their engagement profiles, which can be deduced from the student behaviours and interactions on the system [1, 2]. Several clustering algorithms have been applied to educational datasets, including K-means, Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), Self-Organizing Maps (SOM), Spectral clustering, Affinity Propagation, etc. Studies have shown the effectiveness of clustering techniques in understanding student engagement. For example, Moubayed [3] utilised K-means clustering to identify distinct engagement patterns in an online course, highlighting the importance of personalising interventions to enhance student motivation and participation. Howlin [4] proposed a repeated fuzzy clustering algorithm for discovering student behaviours or outliers. Another study [5] divides students into different types according to hierarchical clustering and uses collaborative filtering AI algorithms for lesson recommendations. This paper uses clustering techniques to analyse a student engagement dataset derived from Moodle logs. The objective is to uncover meaningful patterns within the data and identify distinct groups of students based on their engagement profiles. These patterns can then provide insights for enhancing instructional strategies and interventions. This research uses the K-means, BIRCH, agglomerative clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and Gaussian Mixture Model (GMM)

algorithms to examine the dataset, compare how well they work, and discuss the results. It first collects the logs from the Moodle platform and then calculates parameters using the CoI framework to create a secondary dataset. Next, clustering techniques are employed to analyse this derived student engagement data, and the groups of students are categorized based on their interactions within the online system. Further analyses are then presented to discuss the best suitable algorithms for such kinds of data and the clusters produced by them. This research contributes to educational data mining and provides valuable insights for educators, instructional designers, and administrators. The findings can be used in adaptive learning environments for evidence-based decision-making, improve student engagement, and ultimately enhance the effectiveness of online learning environments.

The remainder of this paper is organized as follows: Section II presents the Literature Review, and Section III presents the methodology for calculating the number of sessions and their engagement and duration. It also details the clustering algorithm performed on the derived parameters. Section IV discusses the results, offers an overview of the work's outcomes, and explains how this reported work advances similar existing work. Finally, section V concludes the paper and presents further research opportunities.

II. Literature Review

Student engagement is a process whereby students are actively involved in their learning. This means that they are not simply absorbing information but are actively thinking about what they are learning, questioning it, applying it and connecting it to their own lives. When students are engaged in their learning, they are more likely to remember what they have learned and be able to apply it in different contexts. Engagement is defined as the level of deliberation, intellectual curiosity, interest, enthusiasm, and passion that learners demonstrate while learning or being instructed, which extends to the level of their progress. Engagement measurement can be conducted in various ways, as mentioned earlier. A detailed comparison of these methods is made in the results and discussion section and summarized in Table A1. Since the log trace analysis is the best method due to its simplicity and reliance on no other additional hardware, a new method in this category of engagement detection techniques is proposed in the current study.

In machine learning, clustering is a technique to group similar data points. It is a way of learning that does not require supervision and helps to discover structure or pattern in a data set that has not been categorised [6]. Clustering techniques are utilised in various domains, including but not limited to machine learning, data mining, pattern recognition, image analysis, information retrieval, and bioinformatics [7]. Using clustering models on educational data has increased over the past decade [8, 9]. It is used primarily for group instances, like students, based on similarity measures to uncover hidden patterns in educational data, such as understanding student achievement [10] and characterising students' learning behaviours [11]. Clustering in a student dataset can assist in identifying groups of students who share similar features or performance patterns [12–15]. This can be beneficial for specific teaching or intervention tactics. If a group of pupils is found to be underperforming in a specific topic, they can receive extra resources or assistance. Likewise, high-achieving kids may receive advanced resources or be transferred to an accelerated curriculum. The dataset and the unique task requirements primarily influence the selection of a clustering algorithm. In the current study, applying clustering techniques to the students' dataset is an attempt to identify patterns and accordingly judge the algorithms that could detect the engagement levels of the students along with related parameters. It describes the application of various clustering algorithms to educational datasets, followed by a comparison to find which one is best suited. This paper discusses the application of the K-means, Agglomerative Hierarchy Clustering algorithm, BIRCH, GMM and DBSCAN that were chosen as they were found to be most common among the studies similar to the current one [2, 16, 17]. The current work relies on three derived parameters that cluster the students to identify the different engaged groups. It differs from other previous works [18–22] that use the Moodle logs. Most of these related works use a sum of clicks or frequency of use of the elements in the LMS or involve the students' socioeconomic, demographic and other backgrounds. In contrast, the parameters used in the current study are the number of sessions, session duration, and session engagement and are simplest to capture and process. The first parameter is calculated by identifying the sessions student-wise. and then sum them up to give the total number of sessions. In the next step, the time of the login and time of logout is used to calculate the duration of each session, and then the duration of all the sessions for a student are added up. The third parameter is the student's engagement, which is calculated based on every interaction that the student has with various events set in the course, as explained in the Data preprocessing subsection of the next section. This last parameter is the major differentiating factor compared to features used in other similar works, as it emphasizes not only the interaction of the student but also the level of each of these interactions within the activity based on the varying levels of cognitive and social involvement of the students as per the CoI framework [23]. Further, no activity on Moodle LMS setup in the course is skipped to calculate the total engagement, as in the case of other related research work [24, 25]. Using this information, which seems vital for defining the involvement of the students in the environment that they are using, this study aims to improve the method of grouping the students according to engagement level. This, in turn, can help build a better framework for detecting engagement and predicting the students' performance in advance.

III. Materials and Methods

In this section, the data collection methods are explained first, and then the preprocessing steps of the data to clean and filter it to create a suitable dataset are explained. This is followed by the primary process of deriving the parameters for the final dataset. These steps are summarized in Fig. A1, given in the Appendix. This figure shows that the data logged on Moodle due to students' interactions are downloaded in CSV file format. The Python program reads these into dataframes to preprocess and label logs with engagement level values. This modified dataset is then copied back into CSV files. These files are then read by another Python

program, which is written to identify each student's sessions and calculate the duration and engagement for each session. It then performs feature extraction, i.e., calculation of the three parameters: number of sessions, total duration, and total session engagement. After this initial data handling process, data is again read by the next set of programs that run the various clustering algorithms and give output in visual and text format for better analysis. These outputs are analyzed and compared to find the best clustering algorithms for this kind of dataset. The clusters formed by these well-performing algorithms are discussed to understand the groups they identify. The basic statistics of the clusters formed by each algorithm is also calculated to discuss the data points' cohesion and the clusters' separation. All these analyses also help to understand which feature is the main contributor to finding the engagement groups. Further, to understand the efficiency of the proposed engagement calculation technique, the engagement values are compared with those calculated manually in the next section.

Table 1. Snapshot of student's Moodle logs

Time	User full name	Affected user	Event context	Component	Event name	Description	Origin	IP address
15-09-2022 14:10	Angela Duncan	-	System	System	User has logged in	The user with id '523' has logged in.	web	150.107.42.184
15-09-2022 14:11	Angela Duncan	Angela Duncan	User: Angela Duncan	System	User password updated	The user with id '523' changed their password.	web	150.107.42.184
15-09-2022 14:11	Angela Duncan	Angela Duncan	User: Angela Duncan	System	Dashboard viewed	The user with id '523' has viewed their dashboard	web	150.107.42.184
06-02-2023 19:38	Angela Duncan	-	System	System	User has logged in	The user with id '523' has logged in.	web	150.107.16.30
06-02-2023 19:38	Angela Duncan		User: Angela Duncan	System	Message viewed	The user with id '523' read a message from the user with id '3'.	web	150.107.16.30
06-02-2023 19:39	Angela Duncan	-	File: Marklist	File	Course module viewed	The user with id '523' viewed the 'resource' activity with course module id '2484'.	web	150.107.16.30
06-02-2023 19:39	Angela Duncan	-	Course: Programming and Problem Solving	System	Course viewed	The user with id '523' viewed the course with id '313'.	web	150.107.16.30
06-02-2023 19:41	Angela Duncan	-	System	System	User logged out	The user with id '523' has logged out.	web	150.107.16.30

Table 2. Student logs after pre-processing and with the calculated engagement level

Time	User full name	Event context	Component	Event name	Level of interaction
2/02/23, 10:04	Angela Duncan	System	System	User has logged in	1
2/02/23, 10:04	Angela Duncan	Course: Programming and Problem Solving	System	Course viewed	1
2/02/23, 10:04	Angela Duncan	User: Angela Duncan	System	Dashboard viewed	2
2/02/23, 10:06	Angela Duncan	System	System	User logged out	0
6/02/23, 19:38	Angela Duncan	System	System	User has logged in	1
6/02/23, 19:39	Angela Duncan	File: Marklist	File	Course module viewed	1
6/02/23, 19:39	Angela Duncan	Course: Programming and Problem Solving	System	Course viewed	1
6/02/23, 19:41	Angela Duncan	System	System	User logged out	0

A. Data Collection

The dataset utilised in the study comprises Moodle LMS logs necessary for the investigation and belongs to the students in their first year of the Master's degree. The students were enrolled in the "Programming and Problem Solving" course on Moodle, and the platform was used for the entire semester for six months. Anonymized data was only used of students who gave consent, despite it being mandatory for all students to use Moodle. Two out of the 74 students did not give informed consent; hence, their data was not included in the study. Participants followed the instructor's directions and completed the activities upon logging into the server. They participated in various activities set up on the course page on Moodle, including reading course material, viewing videos, and completing surveys like quizzes, polls, and homework on the course pages. Once a participant logged in, all interactions on the pages were saved in the database. Subsequently, the records of this transaction were acquired and anonymized for analysis. The university's Institutional Human Ethics Committee (IHEC) approved the study proposal for ethical consideration. This raw dataset (anonymized), downloaded from the Moodle server, is shown in Table 1.

B. Data Preprocessing

The student's actions were recorded in the Moodle system, and the logs for each student in the course were retrieved from the server. All the data is processed using algorithms implemented in Python programming language, and the data itself is extracted or exported from/to CSV files. The files were retrieved into a unified Python data frame (a data structure available in the Python library to store different kinds of data), merging all rows from the individual student's files (as shown in Table 1). The data preprocessing on this dataset is divided into two significant steps. In the first step, the data is cleaned, and then levels of engagement for each log are calculated, while the second step involves identifying the sessions of the students.

1) Step 1: Data cleaning and calculating engagement level for each log

The downloaded dataset required cleaning to ensure it was appropriate for generating meaningful interpretations. The subsequent actions, therefore, involved removing the unnecessary columns such as "Description" and "Origin". Some rows were also eliminated since they were deemed redundant, including those related to the admin user, the guest

user or the faculty member who may have been impacted by the user but did not exhibit any direct user activity. The dataset also underwent various sorting and filtering processes to prepare it for further analysis. The records were initially arranged based on the time stamp to organize them chronologically, followed by sorting them by student names to process all the student records together in batches. An essential part of preparing the data for analysis in this study involves assigning the engagement level to each student's log. It was accomplished using the approach outlined in [26] and referenced in [27] with a few adjustments. These studies discuss calculating specific parameters known as "indicators" for each core activity. This is based on the concepts of "Cognitive depth" and "Social breadth" from the CoI model and is used to predict whether the students are at risk of dropping out and not as such for engagement detection. To understand this labelling process, refer to the flowchart in Fig. 1, which shows how the interactions are labelled for the forum activity. The level of student engagement with the activities on Moodle is influenced by the nature of interactions, as depicted in this figure. If the log indicates that the person is viewing the activity, the log will be given a value of 1. When the student clicks and views the topic of discussion, the corresponding log will be assigned a level 2. When a student responds to a discussion post or creates a new one, a value of 3 will be assigned to that activity. When a new post option is chosen, and a discussion is created underneath it, the log will be assigned a value of 4. After creating a post or selecting a reply post, when content is added to the forum under that post, a value of 5 will be assigned to the log. This approach categorizes all activity logs according to the student's level of interactions. This labelling process is implemented as an algorithm to identify rows based on the component name (activity) and event_name (type of interaction) and then assign values according to the activity and interaction. This allocation is noted by adding a column in the dataset labelled 'Level of interaction'. Table 2 displays a snapshot of the student's logs with the calculated interaction values. It shows how the dataset (in Table 1) gets modified after the preprocessing steps and calculation of the interaction level. Similarly, the Python program performs the labelling for every event of every activity the student has carried out based on the algorithm implemented for each activity, as depicted in the flowchart in Fig. 1.

2) Step 2: Identifying student-wise session details

The next algorithm identifies each student's sessions and then calculates their duration. For this, the output from the step 1 algorithm is fed into it. It then generates a dataframe with details on each student's session length for every day of the course, as shown in Table 3. A student may attend one or more sessions, and accordingly, time-stamped logs for these sessions will be included in the dataset. The "event_name" column (refer Table 2) is examined to identify specific sessions by analyzing entries for each user and each date. A session starts with a login string and ends with a logout string. If the logout string is absent, it means that the user may have suddenly ended the session, perhaps by shutting the browser instead of correctly ending it by clicking the logout button. In such a situation, the rows are scrutinised sequentially as usual; however, when another login string instead of a logoutstring is encountered, the row before the next event_name with the login string is considered the end of the session. Using this technique, every session is recognized and annotated with a session id, "Sid", specific to every date. Since a student may log in and leave several times in a single day, the session id assigned is used to count the number of sessions per day that restarts for every new date. The maximum Sid for each date thus gives the total number of sessions the student had for that day. The duration of each session in minutes is then determined using the timestamp of the rows. The marked engagement levels of the rows are also added to calculate the total involvement of the learner for each session. As a result, as shown in Table 3, dataframe is created with the session id, their duration, and the session-wise involvement for each student on each day. This data is then used to create the cumulative values for the input of the clustering algorithms.

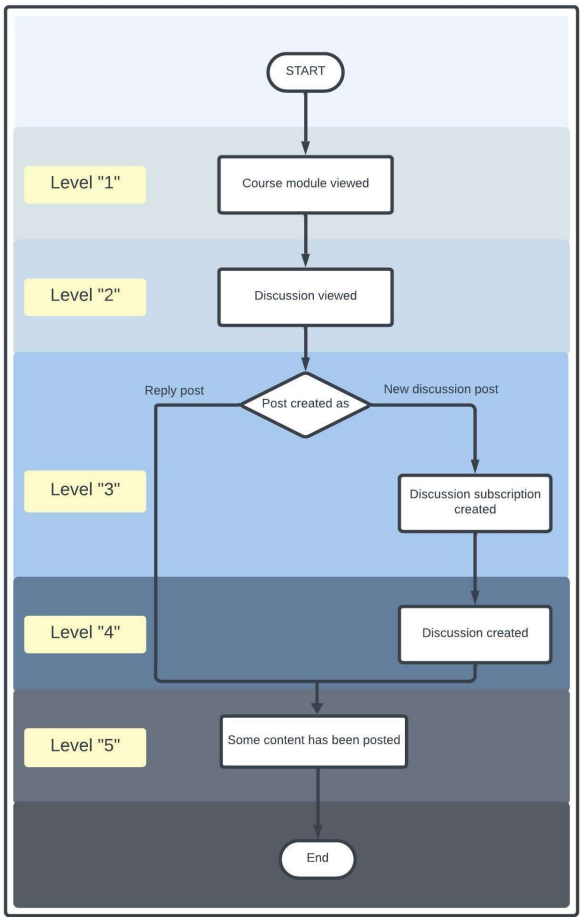

Fig. 1. Flowchart for calculating engagement value for each step of forum activity.

Table 3. Dataset with user-wise session details

user	dates	Sid	SessEng	SessDur
Angela Duncan	02-02-2023	1	2	3
Angela Duncan	16-02-2023	1	3	2
Angela Duncan	19-02-2023	1	20	34
Angela Duncan	19-02-2023	2	5	4
Angela Duncan	19-02-2023	3	56	131
Angela Duncan	20-02-2023	1	56	98
Angela Duncan	12-03-2023	1	50	8
Angela Duncan	22-03-2023	2	38	11
Angela Duncan	23-03-2023	1	89	41
Angela Duncan	23-03-2023	2	28	20

C. Feature Engineering

The final dataset subjected to the various clustering algorithms has three primary features per student: the total number of sessions for the course, the total duration, and the

total engagement value across these sessions. All these parameter values are calculated from the dataframe given in Table 3. The total sessions for each student are calculated by adding the maximum Sid for each date. The total session duration is calculated by adding the duration of all the student's sessions during the course and converting it into total hours. Similarly, the total engagement is calculated by adding all the student's session engagement values. Table 4 shows a few rows of this derived dataset for some students.

D. Data Processing

Clustering, a conventional machine learning technique, is crucial in data analysis. Classifying items based on apparent similarity is fundamental to many scientific disciplines and is a crucial method of comprehension and acquiring knowledge. This analysis systematically examines strategies and techniques for categorising items into groups. A series of measurements or interactions with other objects can define an item. The clustering technique does not use category labels to assign prior identification to items. The lack of category labels distinguishes it from discriminant analysis, pattern recognition, and decision analysis. Its goal is to identify a suitable and accurate arrangement of the data rather than creating guidelines for classifying future data. Clustering methods are designed to identify patterns within the data [28–30]. Many clustering algorithms require a predetermined number of clusters. Identifying the ideal number of clusters can be challenging, mainly when working with a dataset with limited prior information. Many partitional clustering algorithms, such as K-means [31] and K-medoids [32], require the cluster number to be specified as an input parameter before training. Hierarchical clustering methods like BIRCH [33] and clustering algorithms utilising fuzzy theory such as FCM [34] and FCS [35] also require a pre-established number of clusters. Determining the ideal number of clusters for the dataset is crucial in these clustering algorithms. Despite limited prior knowledge about a dataset's features, several approaches are still available to assess the likely optimal number of clusters. The Elbow method is the oldest visual technique to estimate the ideal number of clusters for a given dataset [36]. The next section explains the process of the elbow method and the result of applying it to the dataset under the current study. The following sections then explain the applications of five main clustering algorithms implemented using Python programming language libraries. A summary of various Python programs developed under this study (that implement the various clustering algorithms) and the libraries used in each of these programs are listed in Table A2, given in the appendix.

Table 4. Final Dataset with student-wise parameters

User	Total Sessions	Total Duration	Total engagement
Angela Duncan	242	90.95	2145
Cameron Banks-Brooks	165	14.08	605
Josephine Hughes	193	42.93	1523
Hugh Graham	192	83.42	2006
Mr Derek Parkinson	210	35.58	1418
Terence Williams	194	38.38	1642
Christine Akhtar	190	17.05	978
Damian Berry	183	30.08	1180

1) Elbow method

The Elbow Method is a popular technique used to determine the optimal number of clusters in a dataset for clustering analysis. It involves evaluating the Within-Cluster Sum of Squares (WCSS) for different values of k, where k represents the number of clusters. The WCSS measures the sum of squared distances between each data point and its nearest cluster centroid. A graph is obtained by plotting the WCSS against the number of clusters. This graph typically exhibits a downward trend as the number of clusters increases. The fundamental concept of the Elbow Method is to identify the "elbow point" on the graph, which is the value of k, where adding more clusters no longer significantly decreases the WCSS. This elbow point represents the optimal number of clusters for the given dataset. This method was implemented with the dataset of the current study as input to identify the ideal value for the number of clusters which serves as a starting point for most of the clustering algorithm. The resulting graph is shown in Fig. 2. As per the graph, it is clear that the optimal point is K = 2, after which there is no significant decrease in the WCSS. Therefore, for all further analysis using clustering algorithms, the number of clusters is set to 2 as the initial value.

2) K means

It is a popular clustering algorithm that groups data points into distinct clusters based on similarity [37]. The algorithm iteratively updates the cluster assignments and centroid positions until convergence. The steps of the K-means algorithm are as follows:

1) Randomly initialize the positions of the k centroids.

$$C = \{1, 2, \dots, k\} \tag{1}$$

2) Assign each data point x to the nearest centroid μ_{ij} using the square root of the Euclidean distance as given in Eq. (2):

$$\min_{\mu_i} \sqrt{\sum_{j=1}^{n} \left(x_j - \mu_{ij}\right)^2} \tag{2}$$

3) Update the centroid positions by calculating the mean of the data points assigned to each cluster as per the Eq. (3):

$$\mu_{ij} = \frac{1}{|C_i|} \sum_{x \in C_i} x_j \tag{3}$$

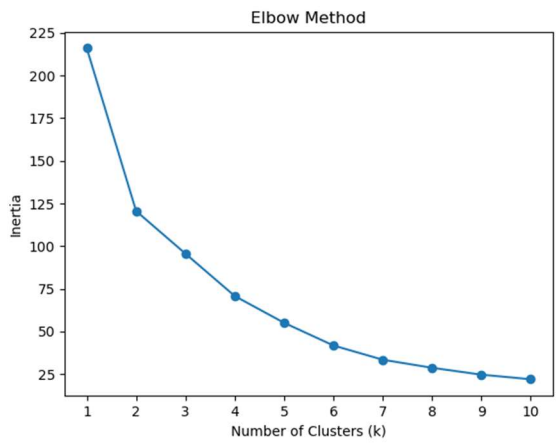

Fig. 2. Output of Elbow method.

4) Repeat steps 2 and 3 until convergence, i.e., until the cluster assignments no longer change significantly.

K-means clustering is a popular unsupervised learning algorithm that can be used for various applications, such as image segmentation, customer segmentation, and anomaly detection [38–40]. K-means aims to minimize the within-cluster sum of squares, which measures the compactness of the clusters. The algorithm converges when the centroids no longer change significantly or when the maximum number of iterations is reached [41]. K-means has several advantages, such as simplicity and efficiency, making it suitable for large datasets. However, it also has some limitations. It assumes that clusters are spherical, equally sized, and have the same density. It can also be sensitive to the initial centroid selection, potentially leading to different outcomes. The output of the K means algorithm implemented in Python with the dataset as input is shown in Fig. 3.

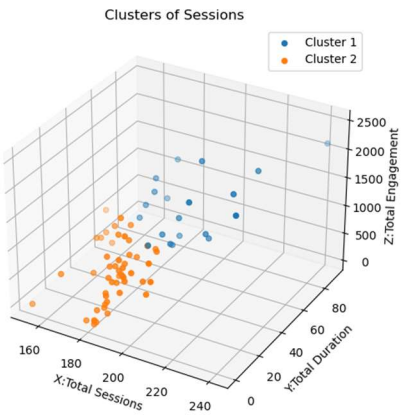

Fig. 3. Clusters formed by K means.

Table 5 shows the mean and std deviation of each cluster as formed with K means. The actual clustered data for each of these is in Tables A3 and A4 in the Appendix. As shown in Table 5, Cluster 1 outperforms Cluster 2 in all metrics: total sessions, total duration, and total engagement, suggesting that Cluster 1 contains more active and engaged users. However, Cluster 1 also shows more significant variability in behavior (higher standard deviations in all categories).

Table 5. Statistics of the clusters formed by the K means algorithm

Cluster	Total Sessions		Total Duration		Total Engagement	
	Mean	Std	Mean	Std	Mean	Std
1	210.75	16.39	30.76	17.79	1788.65	390.74
2	168.35	52.43	12.51	11.34	712.87	370.45

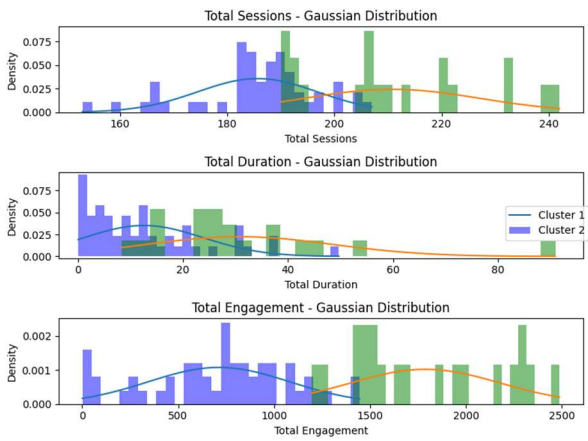

Fig. 4. K- Means: Gaussian distributions of the three parameters.

Visualising the Gaussian distributions, as shown in Fig. 4, helps to understand the contribution of each parameter in forming these clusters. It shows that the third parameter is the best indicator for the clusters, as evident from the overlap between the clusters in the figure when compared across the three parameters. The Total Duration parameters values have the most significant overlap and are the least of the contributors towards these cluster formations.

3) Agglomerative hierarchy clustering algorithm

Agglomerative Clustering belongs to the hierarchical clustering family. This algorithm starts with each data point representing its own cluster and then gradually merges clusters until a termination condition is met [42, 43]. The Agglomerative Clustering algorithm operates in the following manner:

1) Initialization: Each data point is treated as a separate cluster.
2) Similarity Measurement: A similarity measure, such as Euclidean distance or cosine similarity, is used to compute the pairwise distance or dissimilarity between each pair of clusters. In the implemented algorithm, Euclidean distance (d) is used to compute the distance matrix D, which represents the pairwise distances between data points, then it can be computed as given in Eq. (4):

$$D_{ij} = \text{distance}(\text{data point } i, \text{data point } j) \quad (4)$$

with the Euclidean distance between two points/clusters (x, y) calculated with Eq. (5):

$$d(x,y) = \sqrt{\sum_{j=1}^{n} \left(x_j - y_i\right)^2} \quad (5)$$

3) Merge Step: The two clusters with the smallest distance, based on anyone out of the linkage criterion, are merged into a single cluster. Single linkage is used in the implemented algorithm. It computes the distance between two clusters as the minimum distance between any two points, one from each cluster, using Eq. (6):

$$\text{Similarity}(\text{cluster } i, \text{cluster } j) = \min_{x \in \text{cluster } i, y \in \text{cluster } j} \{d(x,y)\} \quad (6)$$

Using this single linkage criterion, the distance matrix D is updated.

4) Repeat: Steps 2 and 3 are repeated iteratively until a termination condition is satisfied. This condition could be the number of clusters or a predetermined threshold value for similarity/dissimilarity.

For the algorithm implemented for this study, cluster merging continues until the specified number of clusters is reached.

The Agglomerative Clustering algorithm follows a bottom-up approach, where individual data points are successively grouped based on similarity. This hierarchical nature enables the algorithm to create a dendrogram or tree-like structure representing the clustering process. The dendrogram can be visually interpreted to understand the relationship between clusters at different levels of similarity. Standard linkage methods used include complete linkage, average linkage, and single linkage, determining how the dissimilarity between two clusters is computed. In this study, we used a single linkage. One key advantage of it is its flexibility in handling different data types. Appropriate distance metrics and linkage methods can accommodate numerical, categorical, and mixed

data types. However, it has a higher computational complexity than other clustering algorithms, especially when dealing with large datasets. The time and space complexity of the algorithm increases as the number of data points grows, making it less suitable for large-scale applications. Despite its computational challenges, it remains a valuable tool in various domains, including image segmentation, social network analysis, and biological clustering. Its ability to capture hierarchical relationships and handle different data types makes it versatile for exploratory data analysis and pattern recognition tasks.

The dendrogram for the current dataset depicting the clusters created at each repeated application of the steps (2 and 3) are shown in Fig. 5. As clear from this diagram, the largest vertical distance that doesn't intersect any other cluster is the one intersecting the blue lines in the dendrogram. It is, therefore, clear that the optimal number of clusters is 2, which are listed in the Appendix under Tables A5 and A6.

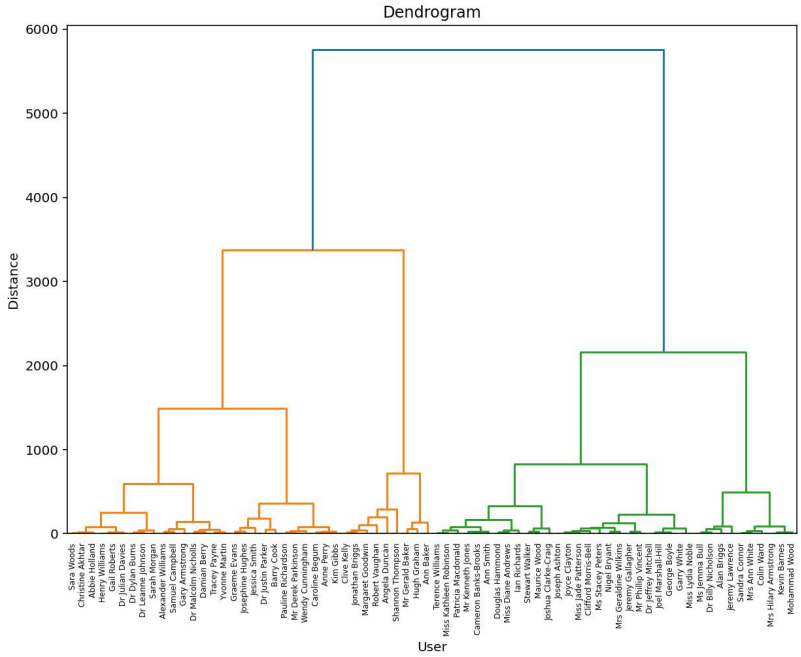

Fig. 5. Dendrogram produced by agglomerative clustering.

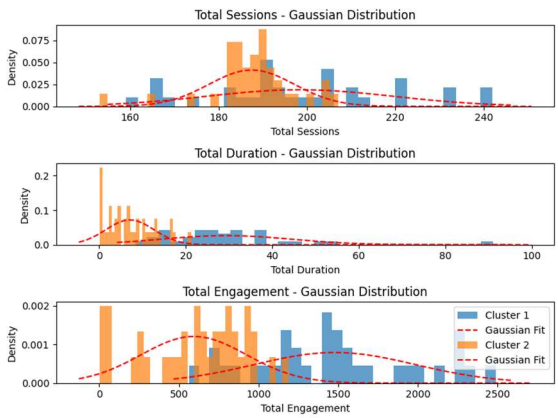

Fig. 6. Agglomerative clustering algorithm: Gaussian distributions of the three parameters for the two clusters generated.

Table 6. Statistics of the clusters formed by the agglomerative clustering algorithm

Cluster	Total Sessions		Total Duration		Total Engagement	
	Mean	Std	Mean	Std	Mean	Std
1	198.38	21.19	29.32	15.13	1478.79	505.27
2	163.79	59.57	7.08	5.54	593.76	331.30

Table 6 displays these clusters' mean and standard deviation for the three parameters. As per this table, although cluster 1 has a higher mean across the three parameters, this difference is significant only in the case of the Total Engagement parameter. Further, the variability of cluster 1 is too high, indicating that the student's behaviour is not too stable. Therefore, this clustering is not very reliable. This is also clear from Fig. 6, which shows the same in graphical form for better understanding. As can be seen, there is large overlap between the clusters; therefore, it is difficult to conclude with certainty regarding the engagement of each of the clusters.

4) Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)

It is an unsupervised machine-learning algorithm designed to efficiently cluster large datasets by creating a hierarchical structure that organizes the data into a tree-like structure called the CF Tree (Clustering Feature Tree) [44]. This tree structure allows for a fast and scalable balanced iterative approach, where it incrementally builds the CF Tree by recursively merging CFs based on clustering, as it maintains a compact summary of the data distribution, referred to as Clustering Features (CFs), rather than storing every individual data point. This algorithm effectively clusters large datasets, making it particularly useful for mining and exploratory data analysis applications. The algorithm implemented from sklearn in the current study has the following steps:

a) CF (Clustering Feature) calculation

$$CF = \left(\sum_{i=1}^{n}\frac{1}{n}, \sum_{i=1}^{n}\frac{1}{n}.\ data_i\ , \sum_{i=1}^{n}\frac{1}{n}.\ data_i^2\right) \quad (7)$$

b) Distance calculation between two CFs

$$\text{dist}(CF_1, CF_2) = \sqrt{\left(\frac{CF_1[2]}{CF_1[1]} - \frac{CF_2[2]}{CF_2[1]}\right)^2 + \left(\frac{CF_1[3]}{CF_1[1]} - \frac{CF_2[3]}{CF_2[1]}\right)^2} \quad (8)$$

c) BIRCH merging criterion

$$\text{dist}(CF_1, CF_2) \leq \text{threshold} \quad (9)$$

d) CF Tree structure

$$\text{CF Tree} = (\text{CF}, \text{CF Tree}_1, \text{CF Tree}_2, \ldots, \text{CF Tree}_k) \quad (10)$$

Eq. (9) is then used to check whether the distance between the clusters is less than the threshold to perform merging. Finally, the tree structure is expanded based on Eq. (9) output. When the dataset was subjected to the code developed on this algorithm, the clusters shown in Fig. 7 were created. Further, the mean and standard deviation for the values of the three parameters of the two clusters are given in Table 7. To understand how these parameters contributed to categorizing the two clusters, the Gaussian distributions were plotted as displayed in Fig. 8.

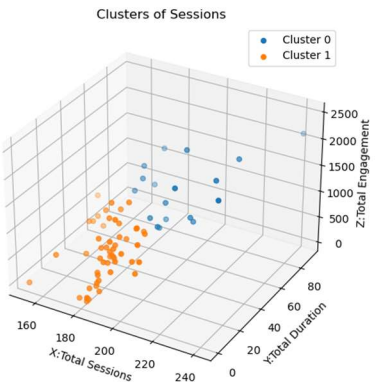

Fig. 7. Clusters created with BIRCH clustering algorithm.

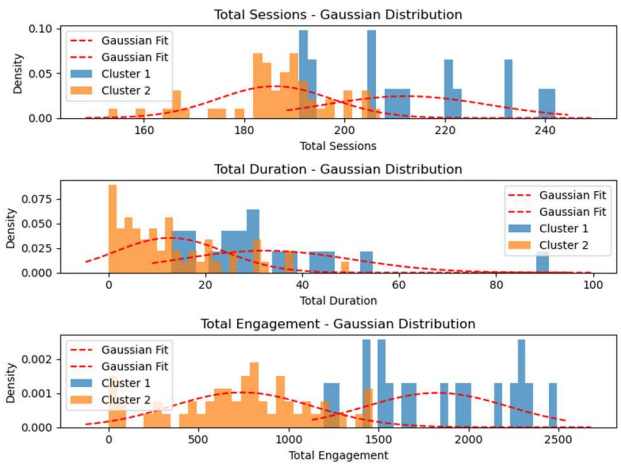

Fig. 8. BIRCH algorithm: Gaussian distributions of the three parameters.

As shown in Fig. 8 and Table 7, students in Cluster 1 are generally more active, spend more time, and engage more than those in Cluster 2. The standard deviations in both clusters are relatively close, although Cluster 1 still has a slightly higher spread. These results are similar to those obtained using K means clustering, as explained in the previous section. It can be concluded that these groupings of the students are better, and these are listed in Tables A7 and A8 in the appendix. This is also evident from the plot of the Gaussian distribution of the three parameters shown in Fig. 8. The total engagement parameter is best suited. In contrast, total duration is the least favourable indicator of the clustered data.

Table 7. Statistics of the clusters formed by the BIRCH clustering algorithm

Cluster	Total Sessions		Total Duration		Total Engagement	
	Mean	Std	Mean	Std	Mean	Std
1	212.28	16.43	32.30	17.88	1824.44	396.02
2	169.41	51.26	12.67	11.28	740.78	390.40

5) Gaussian Mixture Model (GMM)

GMM is a statistical model used to represent complex data distributions. It is a probabilistic model that assumes the data is generated from a mixture of multiple Gaussian distributions, hence the name "Gaussian mixture" [45]. In a GMM, each component represents a Gaussian distribution with its mean and covariance matrix. The mixture model combines these individual Gaussian components to describe the overall distribution of the data. Each component is associated with a weight representing its contribution to the overall distribution. It assumes that the data points are generated by selecting one of the Gaussian components according to their weights and then generating the actual data point from the selected Gaussian component. The model parameters, such as the means, covariances, and weights, can be estimated using maximum likelihood estimation or expectation-maximization algorithms. In the implementation of the current study, the expectation-maximization algorithm is used. GMMs can also be used for density estimation, where they estimate the underlying probability density function of the data. This makes them useful in various fields, such as image processing, speech recognition, and anomaly detection. Despite their effectiveness, GMMs have limitations. They assume that the data distribution is a mixture of Gaussians, which may not always be accurate for complex data. Additionally, estimating the parameters of a GMM can be computationally expensive, especially for high-dimensional data. The models provide a flexible and powerful framework for modelling complex data distributions, clustering, and density estimation, with applications in various data analysis and machine learning fields. The steps of the algorithm are summarized next.

Expectation-Maximization Algorithm for Gaussian Mixture Model:

1) Initialization: Initialize the parameters of the Gaussian mixture model:
   Initialize the means μk, covariances Σk, and weights πk for each component k.
2) E-Step (Expectation step): Compute the responsibilities $\gamma(z_{nk})$ for each data point x_n and component k as shown in Eq. (11):

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n \vee \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n \vee \mu_j, \Sigma_j)} \quad (11)$$

3) M-Step (Maximization step): Update the model parameters using the Eqs. (12-15):
   Update the weights πk:

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma\,(z_{nk}) \quad (12)$$

Update the weights π_k:

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma\,(z_{nk}) \quad (13)$$

Update the means μ_k:

$$\mu_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma\,(z_{nk})x_n \tag{14}$$

Update the covariances Σ_k:

$$\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma\,(z_{nk})(x_n-\mu_k)(x_n-\mu_k)^{\top}$$
$$\text{where } N_k = \sum_{n=1}^{N}\gamma\,(z_{nk}). \tag{15}$$

4) Repeat

Repeat the E-Step and M-Step until convergence criteria are met (e.g., maximum number of iterations or small change in the log-likelihood). In the implemented code, the number of clusters is set to 2, and the tol, the tolerance for convergence parameter in GMM's implementation in sklearn is set to a default value of 1e-3.

Upon application of this algorithm to the dataset of the current study, the clusters as shown in Fig. 9, were generated, while the data split is shown in Tables A9 and A10, given in the appendix. Table 8 summarizes the mean and std deviations for these two clusters along the three parameters.

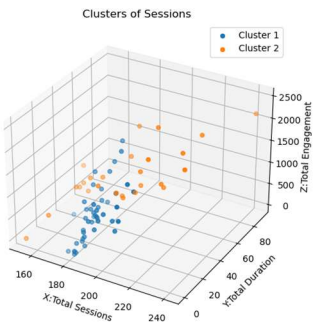

Fig. 9. Clusters created with the GMM clustering algorithm.

Table 8. Statistics of the clusters formed by GMM clustering algorithm

Cluster	Total Sessions		Total Duration		Total Engagement	
	Mean	Std	Mean	Std	Mean	Std
1	173.33	51.18	10.75	8.85	803.02	480.26
2	192.15	41.69	29.66	17.94	1380.88	643.00

The two clusters generated by the GMM algorithm and the respective parameters are plotted in Fig. 10 to show the distribution, mean and standard deviation in graphical form.

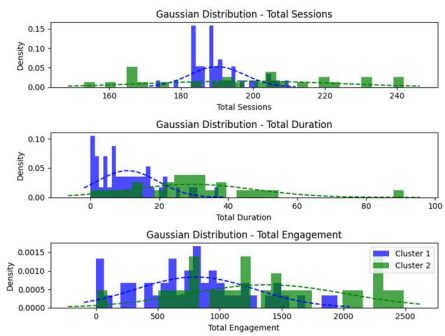

Fig. 10. GMM algorithm: Gaussian distributions for the three parameters.

It is evident from Table 8 that the two clusters are not well-formed as the means (of the parameters) of the two clusters are close to each other while the spread, especially that of cluster 2, is relatively high. Therefore, it could be concluded that the GMM clustering algorithm may not be suitable for the data in the current dataset.

6) Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

It is a popular density-based clustering algorithm used to discover clusters in datasets based on their density distribution [46]. It is beneficial when dealing with data containing clusters of arbitrary shapes and sizes, as it does not make assumptions about the number of clusters or their shapes. It depends on the intuitive concept of "clusters" and "noise". The central idea is that the neighbourhood of a given radius must contain a minimum number of points for each point in a cluster. DBSCAN operates by defining clusters as dense regions of data points separated by areas of lower density. It does not require the number of clusters to be specified beforehand. The algorithm randomly selects an unvisited data point and retrieves its neighboring points within a specified distance, called epsilon (ε). If the number of points in the neighborhood exceeds a predefined threshold, called the minimum number of points (MinPts), the selected point is considered a core point. Core points are central to the formation of clusters. DBSCAN then expands clusters by directly connecting density-reachable points. A data point is considered density-reachable from another core point if it falls within its epsilon neighborhood. A series of core points can also reach density-reachable points. This process continues until no more density-reachable points can be added to a cluster. Points that do not belong to any cluster and do not meet the criteria to be considered core points are labelled as noise points or outliers. DBSCAN can discover clusters of arbitrary shapes and handle noisy data. It also does not require specifying the number of clusters in advance, making it useful for exploratory data analysis. However, DBSCAN's performance can be sensitive to the selection of parameters, such as epsilon (ε) and the minimum number of points (MinPts). Choosing appropriate parameter values is crucial to obtaining meaningful clusters. The steps of the algorithm are given below:

Algorithm: DBSCAN

Input: Dataset D with n data points, Epsilon (ε), Minimum number of samples (MinPts)

Output: Clusters, Noise points

1. Initialize all data points as unvisited.
2. function DBSCAN (D, ε, MinPts):

a) *Initialize an empty list "clusters" to store the clusters.*

b) ***for*** *each unvisited point p in D:*

(1) Mark p as visited.

(2) **if** the number of neighboring points within distance ε (including p) is less than MinPts:

(a) *Mark p as noise.*

(3) **else**:

(a) *Create a new cluster C and add p to C.*

(b) *Expand the cluster C by adding neighboring points within distance ε (including p) to C.*

(c) *Add C to the list of clusters – "clusters".*

c) *return clusters and the noise points.*

The distance between the two points is measured as the Euclidean distance in the implemented version of the

algorithm. Fig. 11 shows the clusters generated from this algorithm with the dataset of the current study.

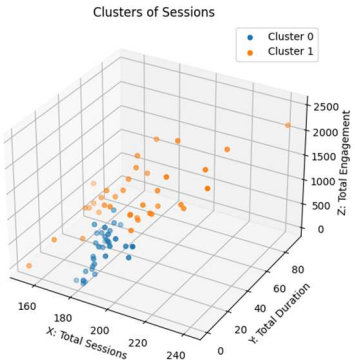

Fig. 11. Clusters created with DBSCAN clustering algorithm.

Table 9 shows the parameter-wise mean and standard deviation of the two clusters. To visualise how each of the parameters separates the clusters (given in Tables A11 and A12), Fig. 12 can be used for further understanding.

Table 9. Statistics of the clusters formed by DBSCAN clustering algorithm

Cluster	Total Sessions		Total Duration		Total Engagement	
	Mean	Std	Mean	Std	Mean	Std
1	167.03	56.93	7.47	5.83	612.40	327.85
2	192.51	35.43	27.14	16.18	1389.41	576.23

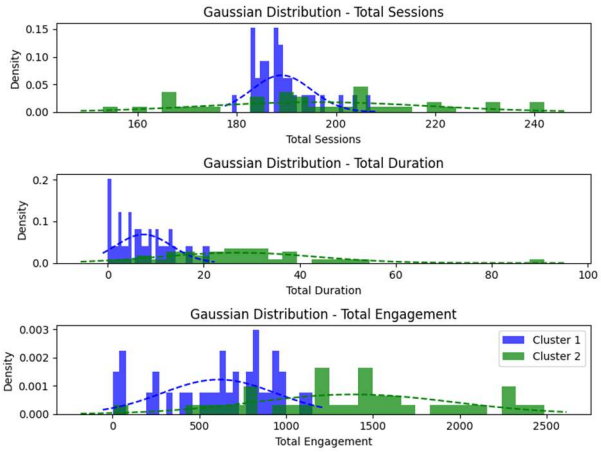

Fig. 12. DBSCAN algorithm: Gaussian distributions for the three parameters

IV. Results and Discussion

A. Proposed Method for Calculating Engagement

This study used a new method of calculating engagement parameters to identify the groups of highly engaged and less engaged students. Engagement measurement can be carried out in several ways. Self-report surveys are extensively utilised, allowing students to select from options or make responses that assess their level of participation [47]. Some recently evolved self-report surveys involve smartphones, which may collect the information regularly using the experience sampling method [48]. However, the reliability of self-reported outcomes is contingent upon several elements beyond the researchers' control, including learners' honesty, willingness to disclose their emotions, and the correctness of their emotional perceptions [49]. Another method involves teacher evaluations or field observations, wherein educators or observers respond. Some disadvantages of these methods involve biasing, false reporting and being unable to scale [50, 51]. Furthermore, they can disrupt the normal learning flow, especially if introduced repeatedly during the learning process [48]. The observation metrics must be properly defined and explained to the observer; otherwise, it could lead to wrong or ambiguous interpretations. For example, sitting quietly and sound behaviour may indicate good involvement; however, they merely mean willingness to adhere to the rules [52]. A distinct device-controlled detection technique incorporates monitoring via external devices and computer vision, such as webcams, which can facilitate eye tracking, capture facial expressions, body posture, and hand motions, and apply algorithms to detect engagement [53]. These are, however, hardware-dependent and must operate continually to monitor actions, requiring the learner to remain tethered to the screen [54]. In addition, they are susceptible to technical issues like face detection failure, incoherent recording frames, etc. [55]. Another approach involving external devices utilises a physiological sensor that captures metrics such as EEG, blood pressure, and heart rate for predictive analysis. These gadgets are hardware-dependent and intrusive, potentially compromising the results, although many attempts are being made to design these sensors to inform the most comfortable wearables [56]. Wearable sensor signals may be influenced by noise, bodily movements, or incorrect sensor positioning [57]. An improved approach that necessitates no additional hardware and is devoid of issues such as false reporting is automatic inference via the logs generated by the learner in an online setting. This work also belongs to this genre of research. However, in most research within this category, not all engagement indicators are employed; for instance, [58] exclusively examines the temporal component, whereas [59] considers only participation duration and frequency. In some instances, just particular activities from the array of options offered in an online environment are utilized. For example, Ramesh [60] exclusively utilize forum activities, while [61] is restricted to an experimental form of content. Further, in most of the existing work, no importance is given either to the social depth of these interactions, where a student goes beyond just interacting with the system and starts communicating with peers or goes to a higher level by sharing his knowledge with the external world or to the cognitive depth where the student goes beyond viewing and is involved in discussing the content or summarizing or applying his knowledge. In this study, on the other hand, it has been proposed that all forms of activity and every user action within it be utilised, regardless of the content type. This is required as the interaction with every activity is essential and contributes to some kind of behavioural engagement, although the level may differ. In addition, for every activity, each allowed interaction is labelled with a value that signifies the depth of the student's interaction.

An additional analysis was conducted to evaluate the efficacy of the proposed strategy. Two raters were engaged to conduct the manual labelling. All activities and interactions pertaining to Moodle were submitted to these evaluators. All participants received identical instructions, and it was guaranteed that the selection of these raters had equivalent familiarity with the Moodle LMS. Only interactions at a participatory level were considered for each activity.

Consequently, additional interactions recorded in Moodle logs or at the instructional levels were omitted. Additionally, the instructions provided to the raters included flowchart samples for activities, which were crafted and elucidated to enable them to apply a consistent technique in determining the depth of all potential interactions within the Moodle LMS. The raters subsequently assigned engagement levels to the events, reflecting their assessment of the interaction depth for each activity. Once completed, the ratings assigned by each inter-rater were input into arrays to conduct inter-rater reliability testing. A Python script utilising the Cohen kappa score class from the sklearn. With these arrays as input, a metrics package was employed to simultaneously assess the degree of concordance between two annotators. This function calculates Cohen's kappa statistics for assessing inter-rater or intra-rater reliability. The additional function employed was ratingtask.multi kappa() from the agreement class of the nltk package, which implements Fleiss' kappa, a statistical metric for evaluating the dependability of agreement among multiple raters. The Fleiss' kappa value obtained was 0.781605, signifying high agreement according to accepted interpretations. Upon establishing the ground truth labels, each log within the dataset stored in the DataFrame was examined separately, and the corresponding level was assigned. This allocation was recorded by incorporating an additional column titled "Man_TotEng" into the previously generated parameter dataset. In the next round of analysis, the engagement values computed by algorithms based on the COI model were compared with those obtained through manual labelling to see if they accurately represent the level of student engagement as observed by faculty. Table 10 presents a comparison of these two methodologies based on numerous statistics.

Table 10. Clustering algorithm performance metrics

Statistic	Manual Labelling	Algorithm-Based
Count	72	72
Mean	1098.08	1011.69
Standard Deviation	608.96	615.60
Min	22.00	1.00
25%	677.50	619.25
Median (50%)	989.00	914.50
75%	1456.25	1415.00
Max	2532.00	2488.00

The correlation value was then calculated as 0.9409, which is very high, indicating a strong positive linear relationship between the two methods. This suggests that both methods produce highly similar results in terms of the overall pattern of engagement values. Also, despite slight differences in means and variability, the two methods largely agree on the relative ranking or behaviour of engagement levels across observations. Further, the Bland Altman plot was created for these values and is shown in Fig. 13. Also known as a Tukey mean-difference plot, it is a method of data plotting used in statistical analysis to assess the agreement between two different measurement methods. It helps assess how well the two methods agree across the range of measurements, identify any bias and the limits of agreement between the methods, and determine whether all the differences in the measurements are within these limits. In the Fig. 13, the x-axis represents the average of the two measurement methods. It ranges from 0 to 2500, covering the engagement scores observed in the data, while the y-axis shows the difference between the two methods. It ranges from −400 to 400. The blue dots represent individual measurements. Each dot corresponds to the mean engagement score (x-axis) and the difference between the two methods (y-axis). The dashed grey line at $y = 0$ represents the line of no difference. If the methods agree perfectly, the points would lie on this line. The two dashed red lines represent the limits of agreement, approximately at $y = 400$ and $y = -400$. These lines indicate the range within which most differences between the methods are expected to lie (mean difference ± 1.96 standard deviations). Therefore, most data points are scattered around the line of no difference, indicating that the two methods generally agree. Further, there is no systematic bias, as the points are spread relatively evenly around the zero line. The spread of points along the y-axis suggests that the differences between the methods are consistent across the range of measurements. The plot visually demonstrates how the two methods of calculating engagement (Manual Labelling and Algorithm-Based) compare. Most differences fall within the limits of agreement, suggesting that while there may be minor discrepancies, the methods are generally consistent and reliable for measuring engagement. This validates the new measurement method, the algorithm-based method devised grounded on the COI model.

Consequently, the suggested approach shows great potential for measuring engagement efficiently and in simple manner compared to similar studies. For instance, in [62], the research uses educational data mining techniques to predict disengagement by analyzing log files from HTML-Tutor, a web-based learning environment. This study applied Bayesian nets, logistic regression, and Simple logistic classification to analyze log files and predict disengagement in e-learning systems. When the overall experiment results of the study are considered, the Attribute Selected Classification using the J48 classifier and Best First search achieved the highest accuracy (90.91) when the output of these methods was verified against the ratings allotted to the student's behaviour based on their log traces. On the other hand, the current method in this study gives higher accuracy while being less computationally complex.

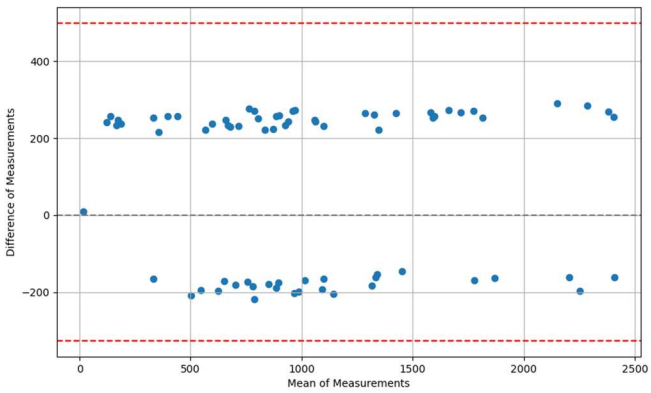

Fig. 13. Bland-Altman Plot for engagement scores.

In another study [63], the authors propose a new automatic multimodal approach that combines and analyzes three modalities: emotions from facial expressions, keyboard keystrokes, and mouse movements. This approach aims to provide real-time, accurate engagement measurements using inexpensive equipment. The accuracy of the proposed

multimodal method for measuring student engagement levels is 95.23%, while the proposed method in the current study has lower accuracy; it is not much less and has a comparatively much simpler technique without reliance on any external devices that need to run continuously to provide the inputs.

Furthermore, the method used to calculate engagement in this work is intensive. It emphasizes not just the quantity of interactions (such as the number of views) but also the quality of engagement, delving into the extent of participation demonstrated by the student (like replying, posting, engaging with peers, the entire class, and so forth). This indicates a promising avenue for developing a relatively straightforward process that enhances the method outlined in the study [64]. In [64], a thorough analysis involving a cohort of undergraduate students demonstrated weak or statistically insignificant correlations between log activities and self-reported engagement, indicating that observable behaviours may not straightforwardly translate into deeper emotional or cognitive experiences. The study emphasized the necessity of considering other underlying factors when interpreting engagement data, emphasizing the complexity of learners' experiences in digital formats. The current study, therefore, extends this work by deriving a dataset that gives importance to the various levels of cognitive and social interactions and achieves a very high accuracy that correlates the log data to the manual labelling of the same.

B. Comparison of the algorithms

Three scores were calculated to compare the performance of the algorithms on the dataset under consideration in the current study. Table 11 shows these scores for all the algorithms.

Table 11. Clustering algorithm performance metrics

Clustering algorithm	Silhouette score	Davies-Bouldin index (DB)	Calinski-Harabasz index (CH)
K-Means	0.431	0.9461	55.3722
Agglomerative Hierarchy	0.3527	1.0797	41.8287
BIRCH	0.4422	0.9286	55.2268
GMM	0.2723	1.7819	16.7389
DBSCAN	0.2924	1.2692	27.0291

1) Silhouette score

It indicates how closely a point resembles the cluster with which it is associated. In other words, for each point, the average distance between that point and the points in the nearest cluster will be calculated minus the average distance between that point and the points in its cluster and then divided by the most significant distance between those distances. The total score is the mean of the points per score. The Silhouette score ranges between −1 and 1, with higher scores indicating more distinct clusters [65].

2) Davies-bouldin index

It is based on the ratio between the within-cluster and between-cluster distances and calculating the average of all clusters. Therefore, it is relatively simple to compute, with a lower score being superior. It can only use the Euclidean distance function because it measures the distance between clusters' centroids [66].

3) Calinski-Harabasz index

This index compares the variance within each cluster to the variance between clusters [67]. This metric is more straightforward to calculate than the Silhouette score but is unbounded. The better the separation is, the higher the score.

Table 11 summarizes the score values for these three parameters of the respective clustering algorithm. Based on this table, it is clear that the two algorithms, K-Means and BIRCH, are the best among the algorithms studied here, with both getting almost the same and the highest Silhouette score and Calinski-Harabasz index, as well as the equal and lowest Davies-Bouldin.

C. Description of the Clusters Produced by Well-Performing Algorithms

As discussed earlier, the two clustering algorithms that give the best cluster output in terms of cluster cohesion and separability on the dataset under consideration in the study are K-Means and BIRCH. Therefore, to understand the behaviour of the dataset, the clusters formed by these two algorithms are discussed. Tables A3 and A4 in the appendix list points in clusters created by the K-Means clustering algorithm, while Tables A7 and A8 show the clusters' data produced by applying the BIRCH algorithm. Cluster 1 of K-Means has 20 students, while Cluster 2 has 52 students. Similarly, cluster 1 of the BIRCH algorithm has 18 students, and cluster 2 has 54 students. Out of the three parameters based on which the clustering is performed, the Total Engagement is found to be the best feature contributing towards the cluster formation (see Figs. 3 and 4 and Table 5 for K-Means and see Figs. 7 and 8 and Table 7 for BIRCH algorithms) followed by Total Sessions. At the same time, Total Duration has the least influence on the clustering. Upon further scrutiny of the data points in respective clusters, it is evident that the cluster 1 students, on average, have higher values for a total number of sessions and very high values for student engagement compared to those in cluster 2 produced by both algorithms. A simple analysis of the values in these clusters (finding minimum and maximum) suggests that threshold values for the grouping of the students into the low and high engagement classes are around 208 and above for the total number of sessions and approximately 1446 for the engagement level for them to be clustered in the high engaged group for similar kinds of data. Further, since the total engagement parameter is found to be the major contributor, it can be concluded that the process of labelling the engagement and then finding the cumulative is effective in categorising a student as engaged or not engaged based on the social and cognitive interactions of the student.

V. Conclusion and Future Work

The current study is part of a larger research project that aims to help calculate the engagement level in online sessions and then predict performance based on these levels. The logs of the Learning Management System are pre-processed and quantified by adding values that represent the engagement level based on the CoI framework. As discussed, this proposed method is much more convenient and efficient because of its reliance solely on the logs generated in the learning environment, which removes the probability of

occurrence of issues like bias and false reporting as well as dependency on invasive or external hardware. Additionally, this approach showed a strong relationship with the manual labelling of these logs carried out according to the provided ratings, and the correlation value significantly exceeded those found in similar studies or was very close to them. Next, unsupervised machine learning approaches, i.e., clustering algorithms, are used to explore the data sets and check if any patterns emerge that can indicate levels of engagement. When subjected to the dataset with these three parameters, all the clustering algorithms, 'Total Sessions', 'Total Duration', and 'Total Engagement'. have produced two clusters (as per the number of clusters fixed based on the elbow method). In most algorithms, the session engagement parameter is crucial in defining clusters' points. This parameter, derived from the distinctive approach introduced in this study for labelling engagement levels, demonstrates the significance of the proposed method for calculating the value. In addition, after comparing the algorithms using the mean and standard deviation of the clusters, the K-means and Birch algorithms were found to be the best clustering techniques on the study's dataset. This is also affirmed by calculating the Davies-Bouldin index (DB), the silhouette coefficient index, and the Calinski-Harabasz index (CH). The output categories of these algorithms can be used to define threshold values that can be used in future to predict performance if the correlation between the engaged students and their performance is established. It can be tested whether the clusters identified as low engagement and high engagement groups in this study have any difference in performance and whether the dataset can be used to build a model that can predict student performance based on which group they belong.

APPENDIX

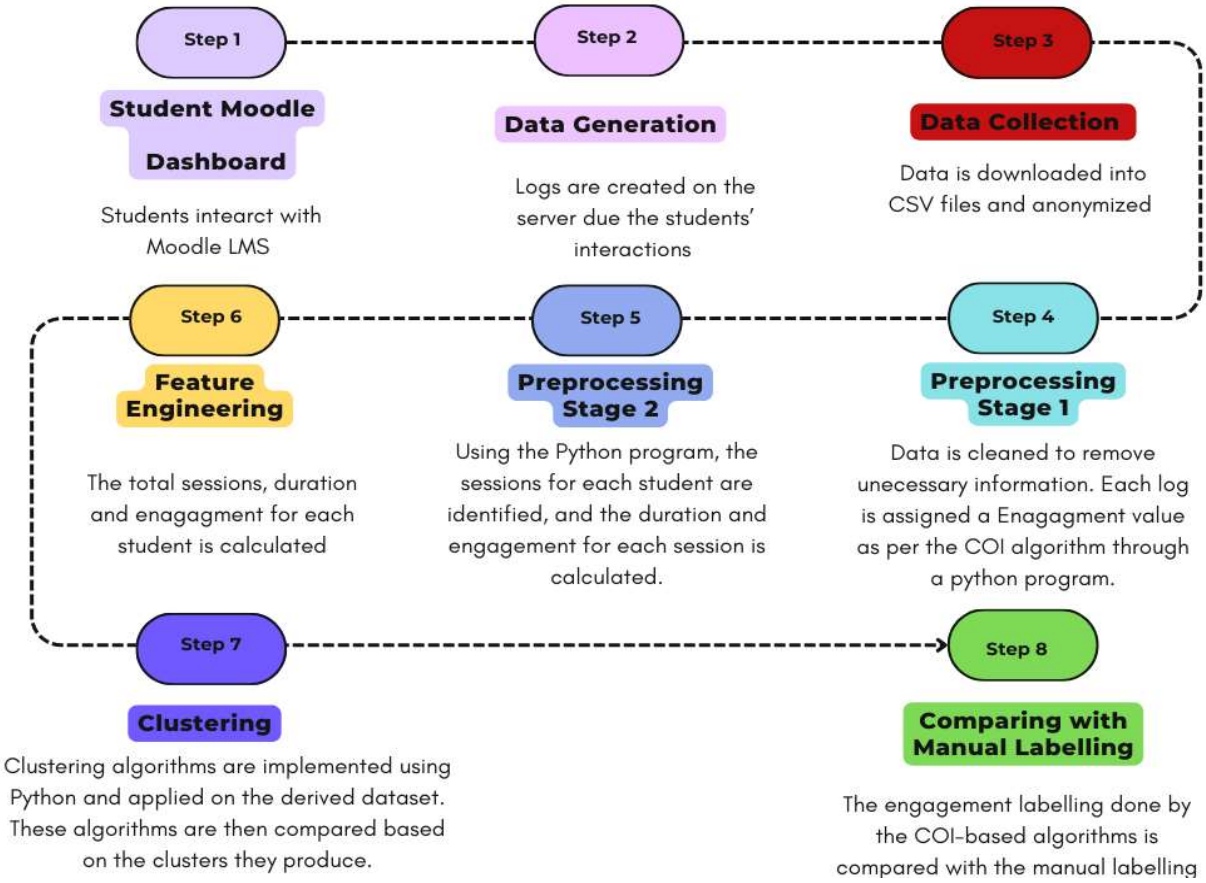

Fig A1. Summary of the steps carried out in the process of calculating the parameters and performing clustering

Table A1. Comparison of the proposed method with other existing methods

Sr. No.	Name of the method	Process used	Disadvantages	Comparison with the proposed method	Related work
1	Self-report Questionnaires	Students are asked questions and have to choose from given options	It heavily relies upon the integrity of learners, their readiness to express emotions, and the accuracy of their emotional feelings. It can cause disruption in the learning process as students may be asked to report at every step.	Does not depend on students to report anything.	[47, 68]
2	Teacher rating/Field Observations	Teachers or other observers in the class report about each student's involvement through questionnaires.	Biasing, false reporting and being unable to scale. It can cause disturbance in class or anxiety in students' minds if	Does not depend on anyone observing or reporting.	[47]

			the observers cannot remain unapparent.		
3	Computer vision based	Capture the movements, especially the facial expressions, and then apply machine learning algorithms to predict the engagement.	Additional external hardware is required that runs throughout and can experience various technical issues like face detection failure, incoherent recording frame, etc.	There is no dependency on any extra additional hardware. It may not face these kinds of technical issues and will work fine as long as logs are properly registered.	[53, 54, 55, 69]
4	Physiological sensor	Using sensors connected to the student's body, various metrics like EEG, blood pressure, respiratory patterns, etc, are captured, and these signals are then fed to algorithms to detect the state.	Hardware dependency, wearing sensors may cause students to become self-aware and behave erratically and, in turn, result in hallucinated data, incorrect sensor positioning, bodily movements may lead to inaccurate signals, and scaling up are some of the disadvantages	No sensors are involved, and therefore, all these issues cannot arise.	[56]
5	Other methods that use log analysis	Students' interactions are tracked and logged for further analysis as they work in a learning environment.	Use only a few logs by applying restrictions on the activities set up or by considering only the numerical measures like the number of times the student has clicked the number of times accessed.	All activities set up in the course are used along with every interaction on these activities. The interactions are also Moreover, in the majority of current research, insufficient emphasis is placed on the social depth of these interactions, wherein a student transcends mere engagement with the system to communicate with peers or the external environment, as well as on the cognitive depth, where the student progresses beyond passive observation to discuss, summarise, or apply their knowledge actively. This study proposes that all forms of activity and every user action be exploited, irrespective of content type. The interaction with each activity is essential and adds to varying degrees of behavioural involvement. Furthermore, each permitted contact for every activity is designated with a value indicating student engagement level. quantified with values as per the depth along the cognitive and social dimensions of CoI.	[58, 59, 60, 61, 70]

Table A2. Python programs and list of libraries used in each one of them

Sr. No.	Name of the algorithm	Name of the program implemented in current study	Purpose	Functions used from library modules	Common functions from the library used across all the programs
1	Kmeans	KmeansDev.py	To read the data, standardize the feature values, perform clustering, create the 3D graph that help to visualize the cluster points, create the Gaussian distribution graph of cluster points depicting distribution for each parameter, each parameter and calculate the scores.	Kmeans from sklearn.cluster	1)StandardScaler from sklearn.preprocessing 2)matplotlib.pyplot 3) norm from scipy.stats 4) Axes3D from mpl_toolkits.mplot3d
2	DBSCAN	DBSCANDev.py		DBSCAN from sklearn.cluster	
3	BIRCH	BIRCHDev.py		Birch from sklearn.cluster	
4	GMM	GMMDev.py		GaussianMixture from sklearn.mixture	
5	Agglomerative Hierarchy Clustering algorithm	AggloDev.py	To read the data, standardize the feature values, perform clustering, create the dendrogram and the 3D graph that help to visualize the cluster points, create the Gaussian distribution graph of cluster points depicting distribution for each parameter, each parameter and calculate the scores	AgglomerativeClustering from sklearn.cluster and dendrogram, linkage from scipy.cluster.hierarchy	
	Note: in addition to above functions from various libraries, all the programs use the silhouette_score, calinski_harabasz_score, davies_bouldin_score functions of sklearn.metrics module in the scikit-learn library to evaluate the performance of models of the various clustering algorithms				

Table A3. Data of the cluster 1 created by K means algorithm

Unnamed: 0	User	Total Sessions	Total Duration	Total Engagement	Cluster
0	Angela Duncan	242	90.95	2145	1
3	Josephine Hughes	193	42.93	1523	1
4	Hugh Graham	192	53.42	2006	1
6	Mr Derek Parkinson	210	15.58	1418	1
13	Yvonne Martin	206	24.18	1195	1
22	Anne Perry	190	25.53	1469	1
23	Graeme Evans	206	16.13	1526	1
29	Robert Vaughan	233	46.18	2350	1
31	Jessica Smith	222	22.92	1582	1
33	Pauline Richardson-Walton	221	27.37	1414	1

39	Caroline Begum	204	8.23	1464	1
40	Ann Baker	194	30.80	1862	1
41	Barry Cook	191	28.88	1639	1
55	Samuel Campbell	206	30.07	1244	1
60	Margaret Goodwin	240	14.32	2247	1
62	Jonathan Briggs	232	27.23	2284	1
64	Dr Justin Parker	209	24.35	1688	1
65	Shannon Thompson	213	35.93	2488	1
67	Clive Kelly	220	12.92	2277	1
69	Mr Gerald Baker	191	37.20	1952	1

Table A4. Data of the cluster 2 created by K means algorithm

Unnamed: 0	User	Total Sessions	Total Duration	Total Engagement	Cluster
1	Dr Billy Nicholson	186	3.30	266	2
2	Cameron Banks-Brooks	165	4.08	605	2
5	Abbie Holland	197	26.18	984	2
7	Miss Lydia Noble	189	11.85	724	2
8	Joyce Clayton	189	10.28	818	2
9	George Boyle	167	32.37	737	2
10	Miss Kathleen Robinson	204	6.33	650	2
11	Terence Williams	194	2.38	642	2
12	Maurice Wood	173	16.38	479	2
14	Mrs Geraldine Wilkins	201	8.92	791	2
15	Joel Marshall-Hill	167	36.58	755	2
16	Christine Akhtar	190	17.05	978	2
17	Dr Julian Davies	207	4.82	935	2
18	Damian Berry	183	30.08	1180	2
19	Ms Stacey Peters	159	49.70	832	2
20	Clifford Morris-Bell	193	6.52	842	2
21	Dr Malcolm Nicholls	188	20.18	1152	2
24	Mrs Ann White	19	0.47	13	2
25	Miss Diane Andrews	188	1.55	534	2
26	Sarah Morgan-Cooke	175	30.53	1068	2
27	Colin Ward-Hart	23	0.07	9	2
28	Mr Phillip Vincent	191	21.20	871	2
30	Patricia Macdonald	185	8.63	677	2
32	Tracey Payne	205	6.97	1188	2
34	Ian Richards	169	37.88	563	2
35	Wendy Cunningham-Black	191	11.95	1407	2
36	Stewart Walker	185	6.13	456	2
37	Dr Leanne Johnson	191	7.12	1086	2
38	Douglas Hammond	185	16.95	550	2
42	Joshua Clarke-Craig	188	2.23	415	2
43	Mr Kenneth Jones	188	9.93	600	2
44	Dr Dylan Burns	201	20.17	1098	2
45	Kevin Barnes	34	0.07	49	2
46	Ms Jemma Bull	184	0.85	247	2
47	Jeremy Gallagher-Adams	190	13.23	894	2
48	Joseph Ashton	186	12.40	823	2
49	Sandra Connor	18	0.03	1	2
50	Gail Roberts	192	8.35	939	2
51	Miss Jade Patterson	195	4.67	808	2
52	Mohammad Wood	38	0.18	69	2
53	Gary Armstrong	200	14.63	1291	2
54	Garry White-Stephens	189	4.83	724	2
56	Dr Jeffrey Mitchell	166	30.50	758	2
57	Mrs Hilary Armstrong	33	0.23	50	2
58	Alan Briggs	188	2.48	204	2
59	Henry Williams	197	6.60	939	2
61	Sara Woods	183	11.02	982	2
63	Jeremy Lawrence	186	3.92	311	2
66	Nigel Bryant-Williamson	183	13.00	770	2
68	Kim Gibbs-Woodward	183	22.32	1446	2
70	Ann Smith	189	13.85	624	2
71	Alexander Williams	184	18.60	1235	2

Table A5. Data of the cluster 1 created by Agglomerative Clustering algorithm

Unnamed: 0	User	Total Sessions	Total Duration	Total Engagement	Cluster
0	Angela Duncan	242	90.95	2145	1
3	Josephine Hughes	193	42.93	1523	1
4	Hugh Graham	192	53.42	2006	1
5	Abbie Holland	197	26.18	984	1
6	Mr Derek Parkinson	210	15.58	1418	1
9	George Boyle	167	32.37	737	1
13	Yvonne Martin	206	24.18	1195	1
15	Joel Marshall-Hill	167	36.58	755	1
18	Damian Berry	183	30.08	1180	1

19	Ms Stacey Peters	159	49.70	832	1
21	Dr Malcolm Nicholls	188	20.18	1152	1
22	Anne Perry	190	25.53	1469	1
23	Graeme Evans	206	16.13	1526	1
26	Sarah Morgan-Cooke	175	30.53	1068	1
29	Robert Vaughan	233	46.18	2350	1
31	Jessica Smith	222	22.92	1582	1
33	Pauline Richardson-Walton	221	27.37	1414	1
34	Ian Richards	169	37.88	563	1
35	Wendy Cunningham-Black	191	11.95	1407	1
39	Caroline Begum	204	8.23	1464	1
40	Ann Baker	194	30.80	1862	1
41	Barry Cook	191	28.88	1639	1
44	Dr Dylan Burns	201	20.17	1098	1
53	Gary Armstrong	200	14.63	1291	1
55	Samuel Campbell	206	30.07	1244	1
56	Dr Jeffrey Mitchell	166	30.50	758	1
60	Margaret Goodwin	240	14.32	2247	1
62	Jonathan Briggs	232	27.23	2284	1
64	Dr Justin Parker	209	24.35	1688	1
65	Shannon Thompson	213	35.93	2488	1
67	Clive Kelly	220	12.92	2277	1
68	Kim Gibbs-Woodward	183	22.32	1446	1
69	Mr Gerald Baker	191	37.20	1952	1
71	Alexander Williams	184	18.60	1235	1

Table A6. Data of the cluster 2 created by agglomerative clustering algorithm

Unnamed: 0	User	Total Sessions	Total Duration	Total Engagement	Cluster
1	Dr Billy Nicholson	186	3.30	266	2
2	Cameron Banks-Brooks	165	4.08	605	2
7	Miss Lydia Noble	189	11.85	724	2
8	Joyce Clayton	189	10.28	818	2
10	Miss Kathleen Robinson	204	6.33	650	2
11	Terence Williams	194	2.38	642	2
12	Maurice Wood	173	16.38	479	2
14	Mrs Geraldine Wilkins	201	8.92	791	2
16	Christine Akhtar	190	17.05	978	2
17	Dr Julian Davies	207	4.82	935	2
20	Clifford Morris-Bell	193	6.52	842	2
24	Mrs Ann White	19	0.47	13	2
25	Miss Diane Andrews	188	1.55	534	2
27	Colin Ward-Hart	23	0.07	9	2
28	Mr Phillip Vincent	191	21.20	871	2
30	Patricia Macdonald	185	8.63	677	2
32	Tracey Payne	205	6.97	1188	2
36	Stewart Walker	185	6.13	456	2
37	Dr Leanne Johnson	191	7.12	1086	2
38	Douglas Hammond	185	16.95	550	2
42	Joshua Clarke-Craig	188	2.23	415	2
43	Mr Kenneth Jones	188	9.93	600	2
45	Kevin Barnes	34	0.07	49	2
46	Ms Jemma Bull	184	0.85	247	2
47	Jeremy Gallagher-Adams	190	13.23	894	2
48	Joseph Ashton	186	12.40	823	2
49	Sandra Connor	18	0.03	1	2
50	Gail Roberts	192	8.35	939	2
51	Miss Jade Patterson	195	4.67	808	2
52	Mohammad Wood	38	0.18	69	2
54	Garry White-Stephens	189	4.83	724	2
57	Mrs Hilary Armstrong	33	0.23	50	2
58	Alan Briggs	188	2.48	204	2
59	Henry Williams	197	6.60	939	2
61	Sara Woods	183	11.02	982	2
63	Jeremy Lawrence	186	3.92	311	2
66	Nigel Bryant-Williamson	183	13.00	770	2
70	Ann Smith	189	13.85	624	2

Table A7. Data of the cluster 1 created by BIRCH algorithm

Unnamed: 0	User	Total Sessions	Total Duration	Total Engagement	Cluster
0	Alexandra Allen	242	90.95	2145	1
3	Josephine Hughes	193	42.93	1523	1
4	Hugh Graham	192	53.42	2006	1
6	Mr Derek Parkinson	210	15.58	1418	1
13	Yvonne Martin	206	24.18	1195	1
23	Graeme Evans	206	16.13	1526	1
29	Robert Vaughan	233	46.18	2350	1
31	Jessica Smith	222	22.92	1582	1

33	Pauline Richardson-Walton	221	27.37	1414	1
40	Ann Baker	194	30.80	1862	1
41	Barry Cook	191	28.88	1639	1
55	Samuel Campbell	206	30.07	1244	1
60	Margaret Goodwin	240	14.32	2247	1
62	Jonathan Briggs	232	27.23	2284	1
64	Dr Justin Parker	209	24.35	1688	1
65	Shannon Thompson	213	35.93	2488	1
67	Clive Kelly	220	12.92	2277	1
69	Mr Gerald Baker	191	37.20	1952	1

Table A8. Data of the cluster 2 created by BIRCH algorithm

Unnamed: 0	User	Total Sessions	Total Duration	Total Engagement	Cluster
1	Dr Billy Nicholson	186	3.30	266	2
2	Cameron Banks-Brooks	165	4.08	605	2
5	Abbie Holland	197	26.18	984	2
7	Miss Lydia Noble	189	11.85	724	2
8	Joyce Clayton	189	10.28	818	2
9	George Boyle	167	32.37	737	2
10	Miss Kathleen Robinson	204	6.33	650	2
11	Terence Williams	194	2.38	642	2
12	Maurice Wood	173	16.38	479	2
14	Mrs Geraldine Wilkins	201	8.92	791	2
15	Joel Marshall-Hill	167	36.58	755	2
16	Christine Akhtar	190	17.05	978	2
17	Dr Julian Davies	207	4.82	935	2
18	Damian Berry	183	30.08	1180	2
19	Ms Stacey Peters	159	49.70	832	2
20	Clifford Morris-Bell	193	6.52	842	2
21	Dr Malcolm Nicholls	188	20.18	1152	2
22	Anne Perry	190	25.53	1469	2
24	Mrs Ann White	19	0.47	13	2
25	Miss Diane Andrews	188	1.55	534	2
26	Sarah Morgan-Cooke	175	30.53	1068	2
27	Colin Ward-Hart	23	0.07	9	2
28	Mr Phillip Vincent	191	21.20	871	2
30	Patricia Macdonald	185	8.63	677	2
32	Tracey Payne	205	6.97	1188	2
34	Ian Richards	169	37.88	563	2
35	Wendy Cunningham-Black	191	11.95	1407	2
36	Stewart Walker	185	6.13	456	2
37	Dr Leanne Johnson	191	7.12	1086	2
38	Douglas Hammond	185	16.95	550	2
39	Caroline Begum	204	8.23	1464	2
42	Joshua Clarke-Craig	188	2.23	415	2
43	Mr Kenneth Jones	188	9.93	600	2
44	Dr Dylan Burns	201	20.17	1098	2
45	Kevin Barnes	34	0.07	49	2
46	Ms Jemma Bull	184	0.85	247	2
47	Jeremy Gallagher-Adams	190	13.23	894	2
48	Joseph Ashton	186	12.40	823	2
49	Sandra Connor	18	0.03	1	2
50	Gail Roberts	192	8.35	939	2
51	Miss Jade Patterson	195	4.67	808	2
52	Mohammad Wood	38	0.18	69	2
53	Gary Armstrong	200	14.63	1291	2
54	Garry White-Stephens	189	4.83	724	2
56	Dr Jeffrey Mitchell	166	30.50	758	2
57	Mrs Hilary Armstrong	33	0.23	50	2
58	Alan Briggs	188	2.48	204	2
59	Henry Williams	197	6.60	939	2
61	Sara Woods	183	11.02	982	2
63	Jeremy Lawrence	186	3.92	311	2
66	Nigel Bryant-Williamson	183	13.00	770	2
68	Kim Gibbs-Woodward	183	22.32	1446	2
70	Ann Smith	189	13.85	624	2
71	Alexander Williams	184	18.60	1235	2

Table A9. Data of the cluster 1 created by GMM algorithm

Unnamed: 0	User	Total Sessions	Total Duration	Total Engagement	Cluster
1	Dr Billy Nicholson	186	3.30	266	1
6	Mr Derek Parkinson	210	15.58	1418	1
7	Miss Lydia Noble	189	11.85	724	1
8	Joyce Clayton	189	10.28	818	1
10	Miss Kathleen Robinson	204	6.33	650	1
11	Terence Williams	194	2.38	642	1
12	Maurice Wood	173	16.38	479	1

14	Mrs Geraldine Wilkins	201	8.92	791	1
16	Christine Akhtar	190	17.05	978	1
17	Dr Julian Davies	207	4.82	935	1
20	Clifford Morris-Bell	193	6.52	842	1
21	Dr Malcolm Nicholls	188	20.18	1152	1
22	Anne Perry	190	25.53	1469	1
23	Graeme Evans	206	16.13	1526	1
24	Mrs Ann White	19	0.47	13	1
25	Miss Diane Andrews	188	1.55	534	1
28	Mr Phillip Vincent	191	21.20	871	1
30	Patricia Macdonald	185	8.63	677	1
32	Tracey Payne	205	6.97	1188	1
36	Stewart Walker	185	6.13	456	1
37	Dr Leanne Johnson	191	7.12	1086	1
38	Douglas Hammond	185	16.95	550	1
40	Ann Baker	194	30.80	1862	1
41	Barry Cook	191	28.88	1639	1
42	Joshua Clarke-Craig	188	2.23	415	1
43	Mr Kenneth Jones	188	9.93	600	1
45	Kevin Barnes	34	0.07	49	1
46	Ms Jemma Bull	184	0.85	247	1
47	Jeremy Gallagher-Adams	190	13.23	894	1
48	Joseph Ashton	186	12.40	823	1
49	Sandra Connor	18	0.03	1	1
50	Gail Roberts	192	8.35	939	1
51	Miss Jade Patterson	195	4.67	808	1
52	Mohammad Wood	38	0.18	69	1
53	Gary Armstrong	200	14.63	1291	1
54	Garry White-Stephens	189	4.83	724	1
57	Mrs Hilary Armstrong	33	0.23	50	1
58	Alan Briggs	188	2.48	204	1
59	Henry Williams	197	6.60	939	1
61	Sara Woods	183	11.02	982	1
63	Jeremy Lawrence	186	3.92	311	1
66	Nigel Bryant-Williamson	183	13.00	770	1
68	Kim Gibbs-Woodward	183	22.32	1446	1
69	Mr Gerald Baker	191	37.20	1952	1
70	Ann Smith	189	13.85	624	1
71	Alexander Williams	184	18.60	1235	1

Table A10. Data of the cluster 2 created by GMM algorithm

Unnamed: 0	User	Total Sessions	Total Duration	Total Engagement	Cluster
0	Alexandra Allen	242	90.95	2145	2
2	Cameron Banks-Brooks	165	4.08	605	2
3	Josephine Hughes	193	42.93	1523	2
4	Hugh Graham	192	53.42	2006	2
5	Abbie Holland	197	26.18	984	2
9	George Boyle	167	32.37	737	2
13	Yvonne Martin	206	24.18	1195	2
15	Joel Marshall-Hill	167	36.58	755	2
18	Damian Berry	183	30.08	1180	2
19	Ms Stacey Peters	159	49.70	832	2
26	Sarah Morgan-Cooke	175	30.53	1068	2
27	Colin Ward-Hart	23	0.07	9	2
29	Robert Vaughan	233	46.18	2350	2
31	Jessica Smith	222	22.92	1582	2
34	Ian Richards	169	37.88	563	2
35	Wendy Cunningham-Black	191	11.95	1407	2
39	Caroline Begum	204	8.23	1464	2
44	Dr Dylan Burns	201	20.17	1098	2
55	Samuel Campbell	206	30.07	1244	2
56	Dr Jeffrey Mitchell	166	30.50	758	2
60	Margaret Goodwin	240	14.32	2247	2
62	Jonathan Briggs	232	27.23	2284	2
64	Dr Justin Parker	209	24.35	1688	2
65	Shannon Thompson	213	35.93	2488	2
67	Clive Kelly	220	12.92	2277	2

Table A11. Data of the cluster 1 created by DBSCAN algorithm

Unnamed: 0	User	Total Sessions	Total Duration	Total Engagement	Cluster
1	Dr Billy Nicholson	186	3.30	266	1
7	Miss Lydia Noble	189	11.85	724	1
8	Joyce Clayton	189	10.28	818	1
10	Miss Kathleen Robinson	204	6.33	650	1
11	Terence Williams	194	2.38	642	1
14	Mrs Geraldine Wilkins	201	8.92	791	1
16	Christine Akhtar	190	17.05	978	1

17	Dr Julian Davies	207	4.82	935	1
20	Clifford Morris-Bell	193	6.52	842	1
21	Dr Malcolm Nicholls	188	20.18	1152	1
24	Mrs Ann White	19	0.47	13	1
25	Miss Diane Andrews	188	1.55	534	1
28	Mr Phillip Vincent	191	21.20	871	1
30	Patricia Macdonald	185	8.63	677	1
36	Stewart Walker	185	6.13	456	1
37	Dr Leanne Johnson	191	7.12	1086	1
38	Douglas Hammond	185	16.95	550	1
42	Joshua Clarke-Craig	188	2.23	415	1
43	Mr Kenneth Jones	188	9.93	600	1
45	Kevin Barnes	34	0.07	49	1
46	Ms Jemma Bull	184	0.85	247	1
47	Jeremy Gallagher-Adams	190	13.23	894	1
48	Joseph Ashton	186	12.40	823	1
49	Sandra Connor	18	0.03	1	1
50	Gail Roberts	192	8.35	939	1
51	Miss Jade Patterson	195	4.67	808	1
52	Mohammad Wood	38	0.18	69	1
54	Garry White-Stephens	189	4.83	724	1
57	Mrs Hilary Armstrong	33	0.23	50	1
58	Alan Briggs	188	2.48	204	1
59	Henry Williams	197	6.60	939	1
61	Sara Woods	183	11.02	982	1
63	Jeremy Lawrence	186	3.92	311	1
66	Nigel Bryant-Williamson	183	13.00	770	1
70	Ann Smith	189	13.85	624	1

Table A12. Data of the cluster 2 created by DBSCAN algorithm

Unnamed: 0	User	Total Sessions	Total Duration	Total Engagement	Cluster
0	Alexandra Allen	242	90.95	2145	2
2	Cameron Banks-Brooks	165	4.08	605	2
3	Josephine Hughes	193	42.93	1523	2
4	Hugh Graham	192	53.42	2006	2
5	Abbie Holland	197	26.18	984	2
6	Mr Derek Parkinson	210	15.58	1418	2
9	George Boyle	167	32.37	737	2
12	Maurice Wood	173	16.38	479	2
13	Yvonne Martin	206	24.18	1195	2
15	Joel Marshall-Hill	167	36.58	755	2
18	Damian Berry	183	30.08	1180	2
19	Ms Stacey Peters	159	49.70	832	2
22	Anne Perry	190	25.53	1469	2
23	Graeme Evans	206	16.13	1526	2
26	Sarah Morgan-Cooke	175	30.53	1068	2
27	Colin Ward-Hart	23	0.07	9	2
29	Robert Vaughan	233	46.18	2350	2
31	Jessica Smith	222	22.92	1582	2
32	Tracey Payne	205	6.97	1188	2
33	Pauline Richardson-Walton	221	27.37	1414	2
34	Ian Richards	169	37.88	563	2
35	Wendy Cunningham-Black	191	11.95	1407	2
39	Caroline Begum	204	8.23	1464	2
40	Ann Baker	194	30.80	1862	2
41	Barry Cook	191	28.88	1639	2
44	Dr Dylan Burns	201	20.17	1098	2
53	Gary Armstrong	200	14.63	1291	2
55	Samuel Campbell	206	30.07	1244	2
56	Dr Jeffrey Mitchell	166	30.50	758	2
60	Margaret Goodwin	240	14.32	2247	2
62	Jonathan Briggs	232	27.23	2284	2
64	Dr Justin Parker	209	24.35	1688	2
65	Shannon Thompson	213	35.93	2488	2
67	Clive Kelly	220	12.92	2277	2
68	Kim Gibbs-Woodward	183	22.32	1446	2
69	Mr Gerald Baker	191	37.20	1952	2
71	Alexander Williams	184	18.60	1235	2

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

VN conducted the experiment and analysis while VK supervised the experiments and helped write the paper; all authors approved the final version.

FUNDING

This publication is an outcome of the R&D work undertaken project under the Visvesvaraya PhD Scheme of the Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India

Corporation.

REFERENCES

[1] R. S. J. d. Baker, "Data mining," *International Encyclopedia of Education (Third Edition)*, pp. 112–118, 2010. https://doi.org/10.1016/B978-0-08-044894-7.01318-X

[2] A. Dutt, M. Ismail, and T. Herawan, "A systematic review on educational data mining," *IEEE Access*, vol. 5, pp. 15991–16005, Jan. 2017.

[3] A. Moubayed, M. Injadat, A. Shami, and H. Lutfiyya, "Student engagement level in an e-learning environment: Clustering using k-means," *American Journal of Distance Education*, vol. 34, pp. 137–156, 2020.

[4] C. Howlin and C. Dziuban, "Detecting outlier behaviors in student progress trajectories using a repeated fuzzy clustering approach," in *Proc. the 12th International Conference on Educational Data Mining, EDM 2019*, Montreal, Canada, 2019.

[5] M. Wang and Z. Lv, "Construction of personalized learning and knowledge system of chemistry speciality via the internet of things and clustering algorithm," *J. Supercomputer*, vol. 78, pp. 10997–11014, May 2022. https://doi.org/10.1007/s11227-022-04315-8

[6] S. Pitafi, T. Anwar, and Z. Sharif, "A taxonomy of machine learning clustering algorithms, challenges, and future realms," *Applied Sciences*, vol. 13, 2023.

[7] A. Ezugwu, A. Ikotun, O. Oyelade, L. Abualigah, J. Agushaka, C. Eke, and A. Akinyelu, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," *Engineering Applications of Artificial Intelligence*, vol. 110, 104743, 2022.

[8] S. Križani´c, "Educational data mining using cluster analysis and decision tree technique: A case study," *International Journal of Engineering Business Management*, vol. 12, 1847979020908675, 2020.

[9] A. Dutt, S. Aghabozrgi, M. Ismail, and H. Mahroeian, "Clustering algorithms applied in educational data mining," *International Journal of Information and Electronics Engineering*, vol. 5, p. 112, 2015.

[10] S. Liu and M. D'Aquin, "Unsupervised learning for understanding student achievement in a distance learning setting," in *Proc. 2017 IEEE Global Engineering Education Conference (EDUCON),* 2017, pp. 1373–1377.

[11] N. Zhang, G. Biswas, and Y. Dong, "Characterizing students' learning behaviors using unsupervised learning methods," in *Proc. Artificial Intelligence in Education: 18th International Conference, AIED 2017*, Wuhan, China, June 28–July 1, 2017, pp. 430–444.

[12] I. Q. Utami, W.-Y. Hwang, and R. A. Ningrum, "Student's behavior clustering based on ubiquitous learning log data using unsupervised machine learning," *Journal of Advanced Technology and Multidiscipline*, vol. 3, no. 1, pp. 13–20, 2024. https://doi.org/10.20473/jatm.v3i1.55572

[13] J. Walsh, "Using cluster analysis to identify procrastination and student learning strategies in a flipped classroom," *The International Journal of Management Education*, vol. 22, 100936, 2024.

[14] F. Qiu, G. Zhang, and X. Sheng, "Predicting students' performance in e-learning using learning process and behaviour data," *Scientific Reports*, vol. 12, 2022.

[15] E. Tuyishimire, W. Mabuto, P. Gatabazi, and S. Bayisingize, "Detecting learning patterns in tertiary education using k-means clustering," *Information*, vol. 13, 2022.

[16] A. M. Nafuri, N. Sani, N. Zainudin, A. Rahman, and M. Aliff, "Clustering analysis for classifying student academic performance in higher education," *Applied Sciences*, vol. 12, p. 9467, 2022.

[17] K. Palani, P. Stynes, and P. Pathak, "Clustering techniques to identify low-engagement student levels," in *Proc. the 13th International Conference on Computer Supported Education*, vol. 2, 2021, pp. 248–257.

[18] C. Kaensar and W. Wongnin, "Analysis and prediction of student performance based on moodle log data using machine learning techniques," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 18, no. 10, pp. 184–203, 2023.

[19] M. M. Tamada, R. Giusti, and J. F. Netto, "Predicting student performance based on logs in moodle LMS," *IEEE Frontiers in Education Conference (FIE)*, pp. 1–8, 2021.

[20] R. Santos and R. Henriques, "Predicting student performance from moodle logs in higher education: A course-agnostic approach," *Education and New Developments 2023,* vol. 2, 2023.

[21] N. Abuzinadah, M. Umer, and A. Ishaq *et al.*, "Role of convolutional features and machine learning for predicting student academic performance from MOODLE data," *PLOS ONE,* vol. 18, 2023.

[22] Y. T. Badal and R. K. Sungkur, "Predictive modelling and analytics of students' grades using machine learning algorithms," *Education and Information Technologies,* vol. 28, pp. 3027–3057, 2022.

[23] R. M. Das and J. V. Madhusudan, "Collaborative learning and learner engagement within the community of inquiry model: A systematic review," *International Journal of Computers in Education*, vol. 6, no. 2, pp. 60–68, 2023.

[24] M. C. S. Manzanares, J. J. Rodríguez-Díez, S. R. Arribas, J. F. Díez-Pastor, and Y. Ji, "Improve teaching with modalities and collaborative groups in an LMS: An analysis of monitoring using visualisation techniques," *Journal of Computing in Higher Education*, vol. 33, no. 3, pp. 747–778, 2021.

[25] D. Y. Liu, J. Froissard, D. Richards, and A. Atif, "An enhanced learning analytics plugin for Moodle," *ASCILITE Publications*, 2015.

[26] E. Dalton and D. Olive. (May 2024). Students at risk of dropping out MoodleDocs. [Online]. Available: https://docs.moodle.org/402/en/Students_at_risk_of_droppin4g75_out

[27] D. Olive, D. Huynh, M. Reynolds, M. Dougiamas, and D. A. Wiese, "Supervised learning framework for learning management systems," in *Proc. the First International Conference on Data Science, e-Learning and Information Systems*, 2018. https://doi.org/10.1145/3279996.3280014

[28] M. Chaudhry, I. Shafi, M. Mahnoor, D. Vargas, E. Thompson, and I. Ashraf, "A systematic literature review on identifying patterns using unsupervised clustering algorithms: A data mining perspective," *Symmetry*, vol. 15, 2023.

[29] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, pp. 165–193, 2015.

[30] A. Jain and R. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Inc., 1988.

[31] A. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, pp. 651–666, 2010.

[32] H. Park and C. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, pp. 3336–3341, 2009.

[33] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM Sigmod Record*, vol. 25, pp. 103–114, 1996.

[34] J. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, pp. 191–203, 1984.

[35] R. Dave and K. Bhaswan, "Adaptive fuzzy c-shells clustering and detection of ellipses," *IEEE Transactions on Neural Networks*, vol. 3, pp. 643–662, 1992.

[36] J. Yang, J. Lee, M. Choi, and Y. Joo, "A new approach to determine the optimal number of clusters based on the gap statistic," *Machine Learning for Networking*, pp. 227–239, 2020.

[37] M. Ahmed, R. Seraj, and S. Islam, "The K-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, p. 1295, 2020.

[38] K. Deeparani and P. Sudhakar, "Efficient image segmentation and implementation of K-means clustering," *Materials Today: Proceedings*, vol. 45, pp. 8076–8079, 2021.

[39] B. Reddy, C. Rishikeshan, V. Dagumati, A. Prasad, and B. Singh, "Customer segmentation analysis using clustering algorithms," *Intelligent Systems*, pp. 353–368, 2024.

[40] M. Lima, B. Zarpelao, L. Sampaio, J. Rodrigues, T. Abrao, and M. Proenca, "Anomaly detection using baseline and K-means clustering," in *Proc. SoftCOM 2010, 18th International Conference on Software, Telecommunications and Computer Networks*, pp. 305–309, 2010.

[41] A. Ikotun, A. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variantsanalysis, and advances in the era of big data," *Information Sciences*, pp. 178–210, 2023.

[42] E. Tokuda, C. Comin, and L. F. Costa, "Revisiting agglomerative clustering," *Physica A: Statistical Mechanics and Its Applications*, vol. 585, 126433, 2022.

[43] K. Sasirekha and P. Baby, "Agglomerative hierarchical clustering algorithm-a," *International Journal of Scientific and Research Publications*, 2013.

[44] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: A new data clustering algorithm and its applications," *Data Mining and Knowledge Discovery*, vol. 1, pp. 141–182, 1997.

[45] D. Reynolds, "Gaussian mixture models," *Encyclopedia of Biometrics*, pp. 659–663, 2009. https://doi.org/10.1007/978-0-387-73003-5125_196

[46] J. Liu, H. Qin, Z. Liu, S. Wang, Q. Zhang, and Z. He, "A density-based spatial clustering of application with noise algorithm and its empirical research," *Highlights in Science, Engineering and Technology*, vol. 7, pp. 174–179, 2022.

[47] J. A. Fredricks and W. McColskey, "The measurement of student engagement: A comparative analysis of various methods and student self-report instruments," *Handbook of Research on Student Engagement*, pp. 763–782, 2012.

[48] G. Sulis, "Exploring the dynamics of engagement in the language classroom: A critical examination of methodological approaches," *Research Methods in Applied Linguistics*, 2024. doi: 10.1016/j.rmal.2024.100162.

[49] S. A. Zhou, P. Hiver, and A. Al-Hoorie, "Measuring L2 engagement: A review of issues and applications," *Student Engagement in the Language Classroom*, pp. 75–98, 2021.

[50] A. W. Sheaffer, C. E. Majeika, A. F. Gilmour, and J. H. Wehby, "Classroom behavior of students with or at risk of EBD: Student gender affects teacher ratings but not direct observations," *Behavioral Disorders*, vol. 46, no. 2, pp. 96–107, 2021. https://doi.org/10.1177/0198742920911651

[51] S. Li, "Measuring cognitive engagement: An overview of measurement instruments and techniques," *International Journal of Psychology and Educational Studies*, vol. 8, pp. 63–76, 2021. 10.52380/ijpes.2021.8.3.239

[52] M. Dewan, M. Murshed, and F. Lin, "Engagement detection in online learning: A review," *Smart Learning Environments*, vol. 6, 2019. doi: 10.1186/s40561-018-0080-z

[53] J. Grafsgaard, J. B. Wiggins, K. E. Boyer, E. N. Wiebe, and J. Lester, "Automatically recognizing facial expression: Predicting engagement and frustration," *Educational Data Mining*, 2013.

[54] B. W. Miller, "Using reading times and eye-movements to measure cognitive engagement," *Educ. Psychol*., vol. 50, no. 1, pp. 31–42, 2015.

[55] S. Saxena, L. K. Fink, and E. B. Lange, "Deep learning models for webcam eye tracking in online experiments," *Behav. Res.*, vol. 56, pp. 3487–3503, 2024. https://doi.org/10.3758/s13428-023-02190-6

[56] M. Bustos-López, N. Cruz-Ramírez, A. Guerra-Hernández, L. N. Sánchez-Morales, N. A. Cruz-Ramos, and G. Alor-Hernández, "Wearables for engagement detection in learning environments: A review," *Biosensors*, vol. 12, no. 7, p. 509, 2022.

[57] N. Thammasan, I. V. Stuldreher, E. Schreuders, M. Giletta, and A.-M. Brouwer, "A usability study of physiological measurement in school using wearable sensors," *Sensors*, vol. 20, no. 18, p. 5380, 2020.

[58] M. Bustos-López, N. Cruz-Ramírez, A. Guerra-Hernández, L. N. Sánchez-Morales, N. A. Cruz-Ramos, and G. Alor-Hernández, "Wearables for engagement detection in learning environments: A review," *Biosensors (Basel)*., vol. 12, no. 7, p. 509, 2022.

[59] L. V. Morris, C. Finnegan, S. S. Wu, "Tracking student behavior, persistence, and achievement in online courses," *The Internet and Higher Education*, vol. 8, no. 3, pp. 221–231, 2005.

[60] A. Ramesh, D. Goldwasser, B. Huang, H. Daume, and L. Getoor, "Uncovering hidden engagement patterns for predicting learner performance in MOOCs," in *Proc. the First ACM Conference on Learning @ Scale Conference*, 2014, 157158.

[61] J. D. Gobert, R. S. Baker, M. B. Wixon, "Operationalizing and detecting disengagement within online science microworlds," *Educ. Psychol.*, vol. 50, no. 1, pp. 43–57, Jan 2015.

[62] E. Haig and S. Weibelzahl, "Disengagement detection in online learning: Validation studies and perspectives," *IEEE Transactions on Learning Technologies*, vol. 4, pp. 114–124, 2011.

[63] K. Altuwairqi, S. K. Jarraya, A. Allinjawi, and M. Hammami, "Student behavior analysis to measure engagement levels in online learning environments," *Signal, Image and Video Processing*, vol. 15, no. 7, pp. 1387–1395, 2021. https://doi.org/10.1007/s11760-021-01869-7

[64] C. R. Henrie, R. G. Bodily, R. Larsen, and C. R. Graham, "Exploring the potential of LMS log data as a proxy measure of student engagement," *Journal of Computing in Higher Education*, vol. 30, pp. 344–362, 2017.

[65] K. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," in *Proc. 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 2020, pp. 747–748.

[66] D. Davies and D. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 224–227, 1979.

[67] S. Saitta, B. Raphael, and I. Smith, "A comprehensive validity index for clustering," *Intelligent Data Analysis*, vol. 12, pp. 529–548, 2008.

[68] B. M. Whitney, Y. Cheng, A. S. Brodersen, and M. R. Hong, "The scale of student engagement in statistics: Development and initial validation," *Journal of Psychoeducational Assessment*, vol. 37, no. 5, pp. 553–565, 2019.

[69] K. Altuwairqi, S. K. Jarraya, and A. Allinjawi *et al.*, "Student behavior analysis to measure engagement levels in online learning environments," *SIViP*, vol. 15, pp. 1387–1395, 2021.

[70] C. Kaensar and W. Wongnin, "Analysis and prediction of student performance based on moodle log data using machine learning techniques," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 18, pp. 184–203, 2023. doi: 10.3991/ijet.v18i10.35841

Copyright © 2025 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).