EasyChair Preprint

№ 5917

Cheating Detection in Online Exams During Covid-19 Pandemic Using Data Mining Techniques

Ali M. Duhaim, Safaa O. Al-Mamory and Mohammed Salih Mahdi

June 27, 2021

Cheating Detection in Online Exams During Covid-19 Pandemic Using Data Mining Techniques

Ali M. Duhaim
Informatics Institute for Postgraduate Studies, Baghdad, Iraq
ms20180516@iips.icci.edu.iq

Safaa O. Al-mamory
University of Information Technology and Communications, Baghdad, Iraq
salmamory@uoitc.edu.iq

Mohammed Salih Mahdi
University of Information Technology and Communications, Baghdad, Iraq
mohammed.salih@uoitc.edu.iq

ABSTRACT

Face-to-face learning has been replaced by e-learning due to the closing of academic institutions in the world during the covid-19 pandemic. Educational institutions faced many challenges in the online platforms and the most important of which was assessing students' performance in the online exams. E-learning has grown significantly every day over the last decade with the growth of the internet and technology. Therefore, an online examination can be beneficial for people to take the exam, but cheating in tests is a common phenomenon around the world. As a consequence, the prevention of cheating can no longer be completely effective. This paper proposed a recommendation system to detect cheating during the online exam using statistical methods, similarity measures, and clustering algorithms by presenting a set of features extracted from the online exam based on Moodle platform. The results show that the proposed online examination system effectively reduces cases of cheating and provides a reliable online exam.

Keywords— *E-learning, Moodle, Online Exam, Cheating Detection, Similarity Measures, Clustering Algorithms*

I. INTRODUCTION

In today's world, e-learning has grown in popularity among academic institutions and organizations. The main benefit of e-learning is that it is accessible to all individuals, regardless of age, place, or time available to learn the contents. The Learning Management System (LMS) is an essential tool in an e-learning system. Many educational institutions use the LMS as a platform to access e-learning materials. In an e-learning environment, students will determine the device for content learning, such as laptop/tablet/mobile. Since the data can be accessed from anywhere, the security of e-learning is the primary objective. Once the students have learned the materials, they must be evaluated by exams. As a result, in an e-learning environment, exams are essential for assessing the learner's performance[1].

Today's online exam is an essential part of e-learning solutions for efficient and equal evaluation of students' results.

The most challenging aspects of e-learning are the design and implementation of online exams. In particular, online examinations are usually performed on e-learning sites without students and teachers being physically present in the same area. This creates some loopholes in online exams in terms of honesty and protection. For instance, in the absence of continuous supervision, an examiner's inspection is highly problematic in the online environment. In addition, online examination environments are susceptible to cheating. It is possible to access many data resources online without any checks or balances from students. Furthermore, maintaining high speed and reliable internet connectivity availability for all students through exams is very difficult to ensure. All of the above issues affected the honesty, protection, and objective existence of online exams [2].

To handle issues surrounding online exams, researchers proposed various methods like biometric methods and

online proctoring to ensure fairness and protection depending on artificial intelligence techniques to prevent cheating during online exams.

This research aims to construct a new model for cheating detection in the online exam based on a reliable dataset and affected features. Also, to create the fairest and effective system for assessing students' performance. In particular, the main contribution of this system is divided into three layers:

1. In the first layer, three online exam features were defined statistically: IP address for each student, the time spent in the exam, and the time late for the exam.
2. In the second layer, the similarity between students' answers was calculated using an overlap similarity algorithm. This layer utilizes the essay question type.
3. The students' answers were divided into similar groups in the third layer using the simple k-means algorithm. The question types used in this layer are (multichoice, true & false, calculated, numerical, multi-answer, and drag & drop).

The rest of the paper is structured as follows. In the second section, we explored the literature review. The third section covers research methodology, including the proposed online exam system and research techniques. In the fourth section, we described the results and evaluation of the system. Finally, the fifth section includes a summary of the system.

II. LITERATURE REVIEW

Educational institutions use the online exam system to improve the quality of education by assessing students' performance in self-paced learning environments. However, despite the importance of the online exam, students engaging in cheating is a widespread phenomenon worldwide [3]. Therefore, in the field of online exams, several academic researchers have been conducted, including continuous authentication, biometrics methods, face-tracking techniques, and other approaches described below:

1. Biometrics Techniques

Biometric authentication is one of the most common techniques for verifying participant identity in online exam environments. This authentication method compares a recorded biometric sample with recent biometrics captured to identify the student. Biometric technologies can be classified into two types: those that involve contact with a scanner (like fingerprints) and those that do not (like eyes). In addition, biometrics typically uses soft characteristics such as (weight, height, age, and gender) and physiological features like (eyes, behavioral factors like mouse movement, signature, and keystroke dynamics). Combining two or more of the above features increases the accuracy of program recognition and is vital to ensure protection [4].

For instance, *Chuang et al.* [5] introduced a method for determining head position and time delay for detecting cheating in the online exam session. They also discussed that a student's head position variation compared to a computer screen has a strong statistical relationship with cheating behavior. Thereby can automatically identify suspicious student activities in the online course. Similarly, *HU et al.* [6] proposed a new method for monitoring the student's abnormal behavior during an online exam, which determines the relationship between the head and mouth of the examinee through a webcam. Experiments have shown that the proposed method was effective for identifying abnormal behavior in the online course.

Moreover, students' strategies for detecting cheating in online exams were discussed. *Bawarith et al.* [7] suggested an e-exam monitoring system to detect and avoid cheating during the exam. The system used continuous authentication of the fingerprint reader and the eye tribe tracker. As a result, the system classified the examinee's status as cheating or non-cheating based on two parameters: the examinee's total time on screen and the number of times the examinee is off-screen. *Mungai et al.* [8] reviewed the significance of keystroke dynamics in keeping security in online exams. The proposed system used a three-stage authentication method, using statistical verification, machine learning, and logical comparison. When an applicant first logs into the system, his typing style is automatically registered, and a template is generated for him. These templates are used as a guide to ensure that the user is authenticated at all times when taking an online exam, based on several parameters, which are: dwell time (time difference between pressing and releasing keys) and flight time (time difference between key release and the next keypress) and typing speed of user for better precision and responsiveness. In a similar study, *Singh and Saurabh* [9] also discussed the keystroke dynamics technique, which requires no pre-registration and can monitor each student's typing behavior during the session. This study ensures that the individual who accesses the resources during the session is the same person who began it.

Prathish et al. [10] proposed an inference system that would assist the instructor in monitoring students during the online exam. They identified the examinee's face based on differences in yaw direction, audio appearance,

and successful window capture. The system was checked in an e-learning environment and effectively achieved in online exam monitoring. In a similar study, *Ketab et al.* [11] presented the development of a more reliable, flexible, and continuous authentication system for online assessments. The system has a continuous user identification using multimodal biometrics to monitor the examiner to ensure that only a valid student takes the exam; a security layer that uses an eye tracker to watch/record student eye movement; and speech recognition to detect unwanted contact. *Mahadi et al.* [12] discussed several techniques and suggested combining (facial recognition and keystroke dynamics) could be the best classifiers in the online course for behavioral biometric authentication. Similarly, *Ghizlane et al.* [3] also suggested a combination of smart cards (to check student's identity) and face recognition technique (for continuous monitoring of a student's webcam) to detect any suspicious behavior during the online exam and avoid any kinds of cheating attempts. *Shdaifat et al.* [13] proposed a model that uses a biometric iris recognition technique in addition to the traditional method of mobile examination login in mobile learning. The suggested model captures iris images randomly, which helps improve the student's authentication during the exam. The study aimed to avoid student impersonation and cheating in mobile exams. Another study implemented by *Garg et al.* [14] suggested a secure system to track students' faces during the exam. Their model was constructed based on deep learning techniques to detect the faces of exam candidates and monitor their behavior to avoid any suspicious practices such as multiple face detection. A recent study conducted by *Vivian et al.* [15] using Deep Neural Network (DNN) to reliably verify the true identity of the student before or during an exam whether on online mode or face to face. The research aims to mitigate examination impersonation by using face-scanning on mobile devices.

2. Video Summarization Techniques

Video summarization applications, also known as video abstraction, used artificial intelligence techniques to detect cheating activities during exams. Students are recorded during the exam using their webcams. If cheating happens, the software will mark the video for analysis by a proctor. Thus, students are monitored, and the time requirements of the supervisors are decreased [4]. Video abstraction is a technique for creating a quick summary of a video, either a set of static images or a sequence of moving images. These ways express the possible cheating case for future evaluation by a human supervisor [16].

For instance, *Cote, et al.* [16] proposed a system based on abnormal student behavior using head pose estimations. Video summaries were produced from irregular behavior sequences observed during evaluation sessions. Results were very promising and indicated to produce real-time warnings for remote monitoring. In addition, an automated method for detecting cheating in online exams has been developed. *Jalali, et al.* [17] monitored exam activities using a webcam that records various images of students. After processing and analyzing, the images were compared with the images of students at different times of the exam. The image activity was considered cheating if the subtracted value exceeds the threshold value. Similarly, *Charan, et al.* [18] implemented an intelligent monitoring system to detect suspicious student activity in the examination hall using a high-density camera to record all of the participants in the session. This study helps identify the students' abnormal behavior, avoiding the presence of a supervisor in the hall and providing evidence of cheating.

3. Other Techniques

A study suggested by *Golden, et al.* [19] paraphrasing question was used to minimize the benefits of online cheating. They challenged students with a verbatim test bank question and a paraphrased question for each topic chosen. Students recorded higher performance on verbatim questions comparing to paraphrasing (80.4% vs. 69.1%). The study showed that they could not quickly answer a paraphrased test bank question since it does not appear online in its original and verbatim form. Thereby, cheating is minimized, academic integrity is preserved, and useful for professors who wish to eliminate the risks of using test banks.

A recent study, *Sangalli, et al.* [20] used the K-means algorithm to detect fraud in online exams based on co-occurring activities and course engagement steps, including students communicating with each other responses or fake accounts students use for the correct answers. As a result, distinguished pairs were identified of actual students who collaborate and others who use fake accounts to get the correct answers.

A recent research study by *Kausar, et al.* [21] proposed a secure e-learning system to avoid security attacks and secure information. The authors proposed a trusted fog server-based safe authentication framework for students and instructors. They also introduced another protocol to set up keys for a specified time, such as a seminar, class, or exam. They emphasized that the proposed system effectively reduces the number of unauthorized students, interaction time for students, authentication, and students' confidence levels.

III. PROPOSED MODEL

This section describes the proposed online examination system, problem assumptions, feature extraction, and techniques used in this paper. The following sections discuss the results and implementation.

1. The Proposed Online Examination System

Previous researches in the area of online exam integrity have several limitations. Some have regularly taken images of each student, while others have employed video cameras to record the students' behavior during exams. However, these systems violate the privacy of students and require fast internet access and powerful software. The primary goal of this research is to use data mining techniques to assess students' answers after the exam. The suggested online examination system is described in (Figure 1).

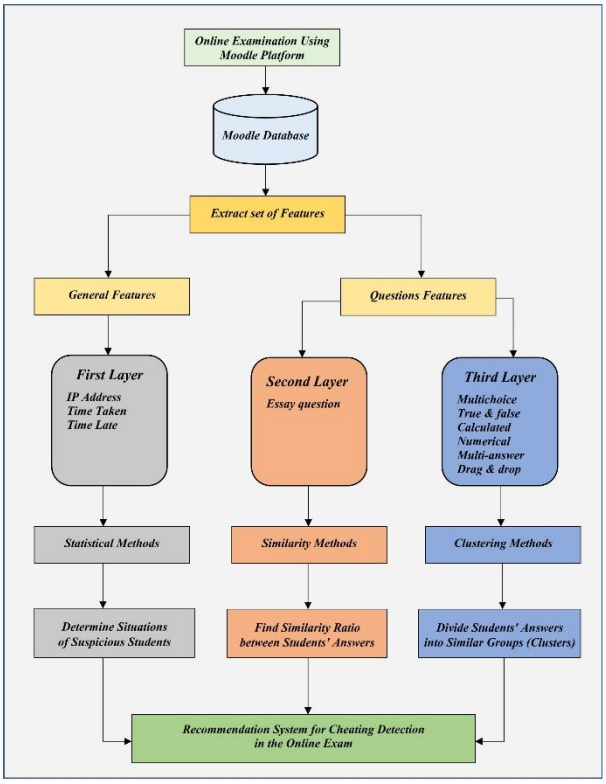

Figure (1): Secured Online Examination System

Initially, the student takes the exam on the Moodle platform, and then exam information is generated and stored in the Moodle database. The entire procedure of our proposed system is organized into three layers. In the first layer, three features derived from the exam (IP address, time taken, and time late) were employed to evaluate the examinee's status using statistical methods. In the second layer, similarity algorithms were utilized to calculate the similarity between students' answers to the essay questions. Clustering algorithms were employed in the third layer to separate students' answers into related groups based on different question types such as multichoice, true & false, calculated, numerical, multi-answer, and drag & drop. Finally, the examiner was provided with a recommendation system for students who cheated in the online exam.

2. Problem assumptions

We considered the following assumptions in this research:

1. The online exam was implemented using the Moodle platform [22].
2. The student must perform his exam alone; otherwise, the proposed system regarded the presence of more than one student in the same location as evidence of cheating.
3. Handwriting questions are not included in our proposed system.
4. When creating an online exam, you can utilize any type of these questions (multichoice, true & false, essay, calculated, numerical, multi-answer, and drag & drop).
5. A recommendation system has been submitted to detect students who cheated in the online exams.

3. Data Preprocessing

A) First layer

Convert time from Unix format to readable date, for example: 1595232387 converted to 08:06:27 AM.

B) Second layer

Some preprocessing tasks are needed in text formatting before comparing essay questions (students' answers). These steps are ordered and stated in the following way:

1. All unwanted symbols are converted to space such as "$", "@", "%", etc.
2. Convert all words from upper case to lower case.
3. All punctuation marks and numbers are removed.
4. All white spaces at the beginning, end, and middle of the document are stripped.
5. The English stop words are removed, which are commonly used terms such as ("the", "an", "a", etc.), since they do not help distinguish between two documents.

C) Third layer

Converting students' answers to an encoding format before implementing the data mining techniques, for example, questions with two responses are converted to 0 and 1, while questions with three responses are transformed to 0, 1, 2, and so on.

4. Features extraction and techniques

This part contains a detailed description of every technique and feature utilized in this research.

A) First layer

To identify cheating situations during an online exam, we use statistical methods based on the following features:

1. **The IP address** is the student's network address, which must be unique for each student. During the exam, most students congregate in one area to exchange answers and assist one another. Thereby, if the students connect to the same network, the system will detect them.
2. **Time taken** is the difference between the finish time and the start time for each student. Several students finish the online exam in a quarter-time given by the examiner, which is against the examination rules because the student cannot leave the exam session while taking the face-to-face exam in such a scenario. The students can share solutions with each other using social media platforms, leading to faster answers, and the exam is done in a quarter of the time.
3. **Time late** is the difference between the start time of the student and the exam's start time. For example, some students are late accessing the online exam at the scheduled time to get the correct answers from others who took the exam.

As a result, our proposed system is considered evidence of cheating when students utilize the same IP address, complete the online exam in a quarter-time, and late for an exam more than ten minutes.

B) Second layer

To calculate the similarity between the answers, we applied similarity measures on the essay questions (as features) in this layer.

An essay question is a test question that requires a written analysis or summary of a specific topic, usually of a defined length. It includes a paragraph, sentence, or short composition. If the ratio of matching between responses is greater than 65%, our proposed system considers it evidence of cheating.

Similarity Measures

The principle of similarity measurement between documents is a fundamental concept in information retrieval and text mining. It is commonly used in Natural Language Processing (NLP) applications like text summarization and machine translation. Data is collected from different sources like online reviews, email, tweets, spreadsheets, and surveys [23]. The primary goal of similarity measurements is to quantify the similarity of two documents or between a document and a query. In other words, the calculation of similarity is a function that measures the degree of similarity between two documents. All similarity measurements fall into the [-1, 1] or [0, 1] range. The minimal similarity is represented by 0 or -1, while absolute similarity is represented by 1 [24]. Three types of similarity algorithms are employed in this layer:

1. Overlap similarity is a measure of how close two sets are. It's determined by dividing the intersection size of two sets by the smaller size of them. If one set is a subset of the other, it is considered a full match [25]. The overlap similarity between A and B is defined as,

$$O(A,B) = \frac{|A \cap B|}{min(|A|,|B|)} \quad (1)$$

The degree of similarity measurement is between 0 and 1. When the two documents are identical, or one of them is a subset of the other, the value is 1; when the two documents are entirely different, the value is 0 [26].

2. Cosine similarity is a measure that specifies how related documents are regardless of their size. Mathematically, it computes the cosine of the angle generated by two vectors projected in multidimensional space [27] . The cosine similarity between A and B is known as,

$$\mathrm{C}(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (2)$$

The value of cosine differs between [-1, 1]. If two documents are identical, their vectors originate in the same direction, creating a slight angle with a cosine value nearby 1. Conversely, when two vectors point in opposite directions from the origin, they form a large angle, and the cosine value is close to -1; thus, the documents are dissimilar, and no similarity is mapped [28], [29].

3. Jaccard similarity compares two sets for similarity. It is defined as the intersection size divided by the union size of two sets [27]. The Jaccard similarity between A and B is referred to as,

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (3)$$

A number between 0 and 1 represents the level of similarity. When the value is 1, two documents are identical; when the value is 0, two documents are dissimilar [28], [29].

C) Third layer

We applied clustering algorithms to separate students' answers into several groups based on the number of k values. The questions types (features) that used in this layer are:

1. **A multiple-choice question (MCQ)** requests the respondent to select one or more options from a limited list. An MCQ includes the correct answer as well as distractors.
2. **A true & false question** is a statement that required a true or false answer. The true & false format can be used in a variety of forms such as "correct" or "incorrect", "yes" or "no" and "agree" or "disagree, etc.
3. **Calculated questions** are specific numerical questions that are based on a formula and use variables or "wild cards" (i.e. {a}, {b}). When the exam is taken, these wild cards are randomly selected from a collection of values.
4. **The numerical question** type needs a number as a response. The values are fixed in the question text.
5. **Questions with multiple answers** allow students to identify more than one choice. When there are multiple correct answers, this form of the question is used.
6. **A drag & drop question** contains a list of two or more potential responses, which can drag to response targets. The goal may be a table, a block, or any other element on the screen.

Details of the Dataset for clustering layer (third layer)

We used 32 examinations from our dataset, specifically final exams from two semesters. The graphic presents the distribution of the different datasets in each exam. The number of attributes and instances are displayed in Figure 2.

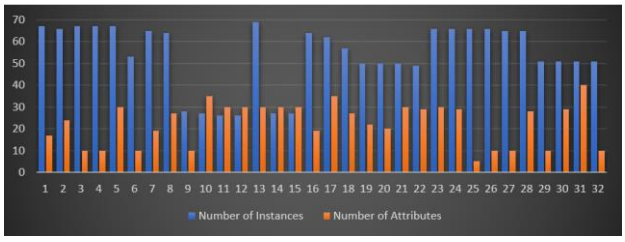

Figure (2): A graphical representation for the number of attributes and instances

The number of instances indicates the total number of students, while the number of attributes represents the total number of questions in each exam.

Clustering Algorithms

Clustering is the process of grouping together similar data objects into clusters. Cluster analysis is used to summarize data, compact it, and find the nearest neighbors efficiently. Different types of clustering are partitional, hierarchical, overlapping, exclusive, fuzzy, complete, and partial. Clustering algorithms are divided into four types: prototype-based clustering, density-based clustering, scalable clustering algorithms, and graph-based clustering. Several important factors must be considered when selecting an effective clustering algorithm, like characteristics of clusters, type of clustering, number of data objects, characteristics of attributes and datasets, cluster description, noise & outliers, and domain-specific issues [30].

Clustering categorizes a set of objects (typically defined as points in multidimensional space) into groups of related objects. Cluster analysis is a valuable component in data analysis. It resembles each other more than patterns from different clusters. The procedure for creating data clusters is shown in Figure 3 [31]:

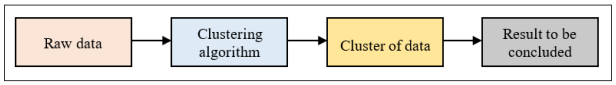

Figure (3): The process of data clustering

In the beginning, we obtain raw data and apply a clustering algorithm to get data clusters. This is the process of using the clustering algorithm to create data clusters. Clustering is commonly used for unsupervised datasets, but it can also be used with supervised datasets. Algorithms play a role in developing a well-designed clustering strategy for a particular problem in clustering. In this layer, three types of clustering algorithms are used:

1. K-Means Clustering Algorithm

k-means algorithm is simple unsupervised learning that works on iterations to group data objects into clusters to solve the well-known clustering problem. The process follows a simple and easy method for classifying a given data set using a specific number of clusters (suppose k clusters). The principal concept is to identify k centers, one for each group. These centers should be strategically placed because different locations produce different results. So, the best choice is to position them far from each other as much as possible. The next step is to associate each point in a dataset with the nearest center. When there are no pending points, the first stage is completed, and an early group age is finished. At this stage, we must re-calculate k new centroids as the barycenter of the clusters generated in the previous step. After obtaining these k new centroids, the same dataset points and the closest new data center have to be linked again. There has been created a loop. This loop means that the k centers change their position step by step until no changes have been made or that the centers no longer shift [32]. Finally, the k-means algorithm aims to

minimize an objective function known as the squared error function [33], which is defined as follows:

$$F = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\|x_i - y_j\|\right)^2 \quad (4)$$

where,

F : represents the objective function.

n : represents the number of clusters.

m: represents the number of instances.

$\|x_i - y_j\|$: represents the Euclidean distance function.

K-means clustering algorithm steps [34]

Let R = (r_1, r_2, …, r_n) be data points set and S = (s_1, s_2, …, s_n) be centers set.

1. The initial cluster centers 'c' is randomly chosen.
2. Compute the distance between all data points and cluster centers.
3. Allocate the information point to the cluster center with the shortest distance between it and all other cluster centers.
4. Use the following formula to re-calculate the new cluster center:

$$\boldsymbol{v}_i = (1/c_i) \sum_{j=1}^{c_i} x_i \quad (5)$$

where 'ci' is the number of data points in i^{th} cluster.

5. Re-calculate the distance of each data point to the new cluster centers.
6. Stop if no data points were reassigned; otherwise, start over at step 3.

2. Hierarchical clustering algorithm

A hierarchical clustering algorithm is one of the most common and simple clustering techniques, which forms a hierarchical cluster arrangement called a dendrogram. The dendrogram tree can be divided into several levels to generate different data clusters. This technique is divided into two types (agglomerative clustering and divisive clustering). The bottom-up approach is used in the agglomerative clustering algorithm. This clustering method assumes each document to be a single cluster, allowing all pairs of clusters to be combined into a single group containing all of the documents. On the other hand, the top-down approach is used in the divisive clustering algorithm—this method of clustering recursively separating the clusters from a single cluster to several groups [34]. Generally, merges and splits are calculated in a greedy manner.

Hierarchical Clustering Algorithm steps [33]

Given a collection of N items for clustering,

1. Begin by assigning each object to its cluster. If you have N items, you will now have N clusters, each including only one item. Let the distances between clusters to match the distances between the objects contained within them.
2. Find the most related (closest) pair of clusters and combine them into a single cluster, resulting in one less cluster.
3. Calculate the distances between each of the old clusters and the new cluster.
4. Steps 2 and 3 can be repeated until all items are grouped into a single N-size cluster.

3. Expectation-Maximization (EM) algorithm

The EM algorithm is an iterative method for determining the maximum likelihood estimates of parameters in mathematical models that depend on unobserved latent variables (variables inferred from the values of other known variables but are not explicitly observable). The Expectation-Maximization iteration alternates between doing an expectation (E) step, which calculates parameters maximizing the expected log-likelihood, and a maximization (M) step, which calculates parameters maximizing the expected log-likelihood found on the E step. In the next E step, these parameter estimates are used to calculate the distribution of the latent variables. EM gives a probability distribution to each case, which indicates the likelihood of it belonging to one of the clusters [35]. This algorithm is the basis of many unsupervised clustering algorithms in machine learning, which is an extension of the k-means algorithm.

Expectation-Maximization Clustering Algorithm steps [36]

1. Select an initial parameter set for the model.
2. E-step: guess the values of the missing data using the dataset's observed available data.
3. M-step: after the expectation (E) step, the complete data generated is used to update the parameters.
4. Repeat steps (2) and (3) until convergence is achieved.

V. RESULTS

The previous section explained the proposed system and every feature & technique that used in this paper. The research results will be discussed in this section.

1. Data collection method

We used a private database in our proposed system provided by an Iraqi university without specifying the university's name for personal reasons. Table 1 shows the basic Moodle statistics of our dataset for the last two years.

Table (1): Moodle statistics for our dataset

Item	Total
Number of courses	180
Number of students	941
Number of quizzes	510
Number of questions	6645
Number of assignments	388
Number of resources	3064

As illustrated in Table 1, the dataset contains 941 participants, 510 exams, 180 courses, 3064 study resources, and 388 assignments for all stages in the first and second semesters. As a result, 32 final exams were used in our proposed system.

2. First layer results

A comprehensive description of the results is offered in Figure 4, which includes the total number of students and the number of students who cheated by (IP address, time taken, and time late).

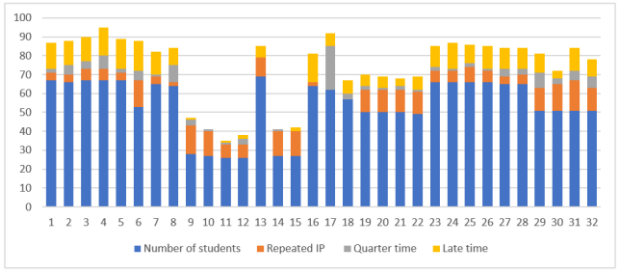

Figure (4): The number of students identified as potential cheaters in the first layer

According to the above graph, which indicates the number of students who cheated in the first layer. The IP address had the highest rate of cheating since most of the students sit in the same location, followed by the time late and time taken.

3. Second layer results

Four sentences with different characteristics were considered for evaluation in the second layer:

1. Set 1 includes two similar documents.
2. Set 2 includes two documents. One of the documents contains a paragraph that exists entirely in the other document.
3. Set 3 includes two different documents on the same subject.
4. Set 4 includes two different documents.

In each of the four sentences, the preprocessing steps in the last section have been used. The Overlap similarity, Cosine similarity, and Jaccard similarity are applied. Figure 5 shows the results of all three measures.

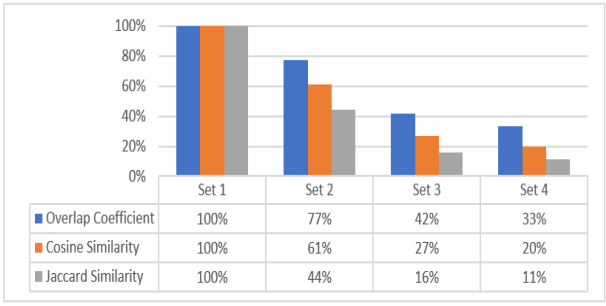

Figure (5): The results of similarity measures

All three measurements have a similarity of 100% in set 1, which contains exactly two identical documents. In sets 2, 3, and 4, the best result is provided by overlap similarity followed by cosine similarity and Jaccard similarity. However, the practical method and similarity measure is based on the characteristics of the experimental data and the work that users plan to do.

The results of the second layer are summarized in Figure (6) below, which include the total number of students and students who cheated in the essay questions.

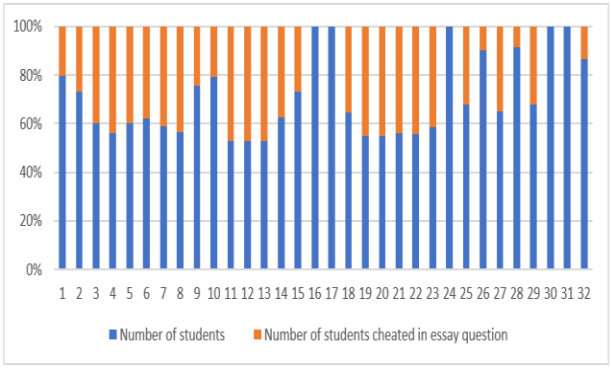

Figure (6): Total number of students and students who cheated in the second layer.

As illustrated in the figure, 27 of 32 exams were cheated by essay questions. As a result, this layer revealed more cheating than the previous one.

4. Third layer results

We compared three clustering algorithms (Simple k-means, Expectation-Maximization, and Hierarchical) based on the number of clusters, the sum of squared error (SSE), cluster instances, log-likelihood, and time is taken to build the model using the Weka (3.8.5) tool. Table 2 displays the results of our experiments while comparing clustering algorithms. The k value (the number of clusters) must be defined for each algorithm.

Table (2) Results in comparison of selected clustering algorithms using the Weka (3.8.5) tool

Name	Exam-1 (number of instances is 66)					Exam-2 (number of instances is 68)				
	Number of Clusters	Cluster Instances	Sum of Squared Error (SSE)	Log-likelihood	Time is taken to build the model (sec)	Number of Clusters	Cluster Instances	Sum of Squared Error (SSE)	Log-likelihood	Time is taken to build the model (sec)
K-Means	2	28 (42%) 38 (58%)	44.750		0	3	30 (44%) 15 (22%) 23 (34%)	32.481		0
EM	2	15 (23%) 51 (77%)		-11.50149	0.01	3	14 (21%) 53 (78%) 1 (1%)		-0.5893	0.11
Hierarchical	2	65 (98%) 1 (2%)			0.01	3	1 (1%) 66 (97%) 1 (1%)			0.02
K-Means	4	18 (27%) 26 (39%) 12 (18%) 10 (15%)	31.072		0	5	21 (31%) 10 (15%) 14 (21%) 8 (12%) 15 (22%)	28.163		0
EM	4	17 (26%) 19 (29%) 20 (30%) 10 (15%)		-4.36442	0.03	5	8 (12%) 48 (71%) 7 (10%) 1 (1%) 4 (6%)		2.2433	0.07
Hierarchical	4	63 (95%) 1 (2%) 1 (2%) 1 (2%)			0.01	5	1 (1%) 64 (94%) 1 (1%) 1 (1%) 1 (1%)			0.01
K-Means	6	14 (21%) 14 (21%) 8 (12%) 10 (15%) 10 (15%) 10 (15%)	26.976		0	7	16 (24%) 11 (16%) 12 (18%) 6 (9%) 11 (16%) 2 (3%) 10 (15%)	24.588		0
EM	6	15 (23%) 8 (12%) 15 (23%) 9 (14%) 9 (14%) 10 (15%)		-4.4077	0.01	7	9 (13%) 27 (40%) 7 (10%) 1 (1%) 11 (16%) 3 (4%) 10 (15%)		5.43905	0.04
Hierarchical	6	60 (91%) 1 (2%) 1 (2%) 1 (2%) 1 (2%) 2 (3%)			0.01	7	1 (1%) 62 (91%) 1 (1%) 1 (1%) 1 (1%) 1 (1%) 1 (1%)			0.01

The best result was obtained from selected clustering algorithms to evaluate our dataset (simple k-means followed by Expectation-Maximization and hierarchical algorithms). The simple k-means algorithm performed best with execution time and clustered instances compared to Expectation Maximization and hierarchical algorithms. Figure 7 shows the results of cluster algorithms compared in terms of time complexity.

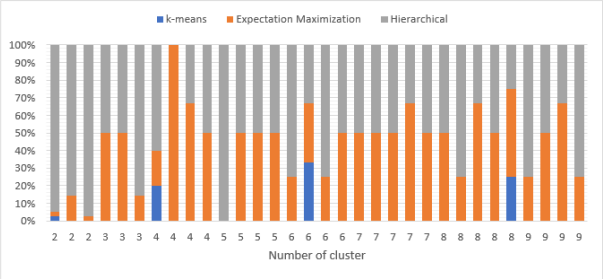

Figure (7): The time taken for the clustering algorithms

The results of the simple K-mean, hierarchical, and EM were compared in terms of time complexity on the 32 exams of our datasets. The k-mean algorithm has the minimum execution time compared to other algorithms.

5. Evaluation Results

Checklist benchmarking can be used to compare suggested and benchmark works on specific points. Several important issues are identified for comparison based on the direction of the proposed topic, which is essential for cheating detection in online exams. Furthermore, checklist benchmarking helps assess how successful the proposed work in comparison with other methods. To conclude, the following four points are demonstrated to show the comparison in the checklists, as shown in Table 4.

1. **First point: IP configuration.** This point indicates whether the study depended on IP addresses or not during the online exam to identify students sitting in the same place by the shared IP address between them.
2. **Second point: Time factor.** This point reflects whether the study used time factor during online exams or not, which is vital in e-learning, such as time spent in the exam, the delay time, the time log in to the exam, etc.
3. **Third point: Scalability participants.** This point reflects the number of students who took the exam and assessed the system's effectiveness based on the number of participants.
4. **Fourth point: Analyzing students' responses.** This point represents students' answers analysis after completing the exam using similarity measures and clustering techniques to assess students' performance.

Table (4): Comparison points between benchmarks and proposed system

Benchmarks	Comparison Points				Total Score	Finding Difference
	IP Configuration	Time Factor	Scalability Participants	Analyzing Students' Responses		
Benchmark#1 [5]	χ	√	√	χ	50%	50%
Benchmark#2 [6]	χ	χ	χ	χ	0%	100%
Benchmark#3 [7]	χ	√	√	χ	50%	50%
Benchmark#4 [8]	χ	√	√	χ	50%	50%
Benchmark#5 [9]	χ	√	√	χ	50%	50%
Benchmark#6 [10]	χ	√	χ	χ	25%	75%
Benchmark#7 [11]	χ	χ	√	χ	25%	75%
Benchmark#8 [12]	χ	χ	χ	χ	0%	100%
Benchmark#9 [3]	χ	χ	χ	χ	0%	100%
Benchmark#10 [13]	χ	χ	χ	χ	0%	100%
Benchmark#11 [14]	χ	χ	χ	χ	0%	100%
Benchmark#12 [15]	χ	χ	χ	χ	0%	100%
Benchmark#13 [16]	χ	χ	χ	χ	0%	100%
Benchmark#14 [17]	χ	χ	√	χ	25%	75%
Benchmark#15 [18]	χ	χ	χ	χ	0%	100%
Benchmark#16 [19]	χ	χ	√	χ	25%	75%
Benchmark#17 [20]	√	√	√	χ	75%	25%
Benchmark#18 [21]	χ	√	√	χ	50%	50%
Proposed study	√	√	√	√	100%	

As shown in Table 4, comparisons are made based on whether or not the compared works addressed the comparison points. For example, the results of the comparison procedure showed that one benchmark study obtained 75% and covered three points (IP configuration, time factor, and scalability participants), five benchmarks studies got 50% and covered only two points for each (time factor and scalability participants), four benchmarks studies obtained 25% and covered only

one point (time factor or scalability participants). Finally, eight benchmarks studies obtained 0% and did not cover any point.

However, the proposed system in this study covered 100% overall, which implies that our results and methodology are fit the limitations of other studies and overcomes them. In addition, the proposed system contributes by providing an all-important points study and presenting an analysis of students' responses that disappeared from their work.

6. Result Implementation

As one of the experiment results of our proposed system, we evaluated the examination status after the exam was finished. We selected a sample from the existing exams for the fourth stage of the second semester. Table 3 displays the system's results.

Table (3): The result implementation of the system

First layer (Statistical layer)				*Second layer (Similarity layer)*	*Third layer (Clustering layer)*
Student ID	IP Address	Time Taken	Time Late	Ratio of similarity	Student' groups
241	Repeated IP (176.10.99.200)	used quarter time of exam: (00:37:20)	on time	Low matching	Group – 1
612	unique IP	used full time of exam	on time	High matching (94%) between 612 and 245	374, 241, 673, 240, 612, 236, 242, 216
635	unique IP	used full time of exam	Time late is: (00:36:38)	High matching (100%) between 635 and 670	
245	unique IP	used full time of exam	on time	High matching (94%) between 245 and 612	
216	unique IP	used full time of exam	on time	High matching (89%) between 216 and 673	
929	unique IP	used full time of exam	on time	High matching (96%) between 929 and 612	
242	Repeated IP (176.10.99.200)	used full time of exam	on time	High matching (96%) between 242 and 671	
670	Repeated IP (185.121.69.40)	used full time of exam	on time	High matching (100%) between 670 and 635	
673	unique IP	used quarter time of exam: (00:25:51)	on time	High matching (91%) between 673 and 245	Group – 2
672	Repeated IP (185.220.103.5)	used full time of exam	Time late is: (00:21:50)	Low matching	635, 672, 222, 670, 227, 671
223	Repeated IP (51.15.82.176)	used full time of exam	on time	High matching (94%) between 223 and 612	
240	Repeated IP (176.10.99.200)	used full time of exam	on time	Low matching	
215	Repeated IP (51.15.82.176)	used quarter time of exam: (00:36:51)	on time	Low matching	
643	unique IP	used full time of exam	Time late is: (00:25:23)	Low matching	
226	unique IP	used full time of exam	on time	Low matching	Group – 3
243	unique IP	used full time of exam	on time	High matching (92%) between 243 and 612	215, 217, 223, 219
220	unique IP	used full time of exam	on time	High matching (94%) between 220 and 242	
221	unique IP	used full time of exam	Time late is: (00:33:07)	Low matching	
374	Repeated IP (176.10.99.200)	used full time of exam	on time	High matching (89%) between 374 and 673	
361	unique IP	used full time of exam	on time	Low matching	
236	unique IP	used full time of exam	on time	High matching (94%) between 236 and 612	Group – 4
217	Repeated IP (51.15.82.176)	used full time of exam	Time late is: (00:17:33)	Low matching	245, 929, 243, 220
671	Repeated IP (185.220.103.5)	used full time of exam	on time	High matching (96%) between 671 and 242	
286	unique IP	used full time of exam	Time late is: (00:27:10)	High matching (90%) between 286 and 220	
222	Repeated IP (185.121.69.40)	used quarter time of exam: (00:31:51)	Time late is: (00:20:38)	Low matching	Group – 5
219	Repeated IP (51.15.82.176)	used full time of exam	on time	High matching (89%) between 219 and 223	643, 226, 221, 361, 286
227	Repeated IP (185.121.69.40)	used full time of exam	on time	Low matching	

This exam was taken by 27 students, and the following results were obtained after implementing our proposed system:

1. In the first layer, 13 duplicate IP addresses were discovered and divided into four groups, meaning that each group of students (241, 374, 240, 242), (670, 222, 227), (672, 671), and (219, 223, 215, 217) sat in the same place to take the exam. In addition, four students (241, 673, 215, 222) completed the exam within the quarter-time limit, whereas seven students (635, 672, 643, 221, 217, 286, 222) were late for taking the exam on time.
2. In the second layer, 16 students had a high matching of answers with other students, while 11 students had a low matching.
3. In the third layer, we divided students' answers into five groups (the number of clusters is five), each group containing several students who had similar responses.
4. Cheating was detected for some students in all three layers, like (242,374), who cheated by using a shared IP address and obtained a high match rate in the second layer, as well as being isolated in the same group in the third layer.
5. Some students did not cheat in the first layer, but cheated in the second layer like (236, 612) and then were isolated in the same group in the third layer.
6. Most of the cheating cases were detected in the second layer; 16 cheats were identified out of 27 students.
7. Some students did not cheat in the first and second layers such as (226,361), but they were grouped in the third layer. This indicates that the students did not cheat or their responses were similar, as showed by the third layer's result. In this instance, the examiner determines whether the students' status is cheating or not.

V. CONCLUSION

Students and educational institutions have paid a lot of attention to e-learning and distance education in the COVID-19 pandemic. E-learning has grown in popularity around the world due to its flexibility, accessibility, and user-friendliness. However, the primary challenge in online education is assessing students in the online exam because cheating in the examination is simple and a significant issue in education and undermining efforts to evaluate a student's performance.

In this paper, we proposed a solution to reduce cheating during online exams. We extract a set of reliable features from the Moodle platform exam using data mining techniques. These features are divided into three layers. In the first layer, we used statistical methods on these features

(IP address, time taken, and time late) to detect cheating in the exam. In the second layer, we applied similarity measurements on essay questions to calculate the ratio of similarity between students' answers. In the third layer, we employed clustering algorithms on these types' questions (multichoice, true & false, calculated, numerical, multi-answer, and drag & drop) to divide students' answers into several related groups. Finally, a recommendation system is presented to assist the examiner in deciding suspicious students' responses.

Some points are identified as a result of our proposed system:

1. Some students cheated at the first layer by sitting in the same location, which was identified by their identical IP address. They also cheated in the second layer, in which they had a high similarity ratio. However, the clustering algorithm grouped them in the third layer.
2. Some students cheated by time (time taken or time late), and they didn't sit in the same location. They also detected in the similarity and cluster algorithms.
3. Some students cheated in the first layer, either by IP address or time and they were also detected in the second and third layers with other students who did not exist in the first layer.
4. The best algorithm we obtained in the third layer was the k-mean algorithm, which required less time to execute and achieved the best clustering instances. As a result, when the number of clusters is large, the SSE is lower, and the clustering instances are better.
5. The second layer had the most significant rate of cheating, followed by the first and third layers.
6. Our proposed system reported that 60% of students cheated in the second course, while 40% cheated in the first course.
7. Students cheat at all educational stages, with the fourth stage cheating is 32%, the third stage is 30%, the second stage is 24%, and the first stage is 14%. Figure 8 shows the number of students who cheated at all academic levels.

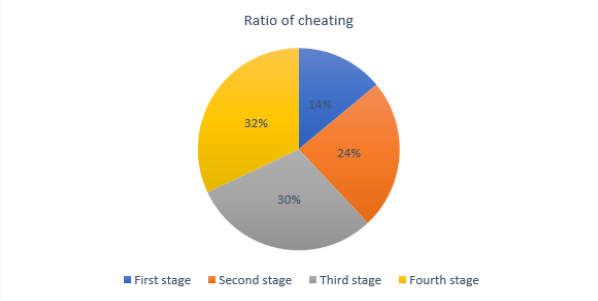

Figure (8): Ratio of cheating for all stages

IV. REFERENCES

[1] J. Deborah L, R. Karthika, P. Vijayakumar, B. S. Rawal, and Y. Wang, "Secure Online Examination System for e-learning," in *2019 IEEE Canadian Conference of Electrical and Computer Engineering, CCECE 2019*, 2019, vol. 2019-Janua, pp. 1–4, doi: 10.1109/CCECE43985.2019.9052408.

[2] A. W. Muzaffar, M. Tahir, M. W. Anwar, Q. Chaudry, S. R. Mir, and Y. Rasheed, "A systematic review of online exams solutions in e-learning: Techniques, tools, and global adoption," *IEEE Access*, vol. 9, pp. 32689–32712, 2021, doi: 10.1109/ACCESS.2021.3060192.

[3] M. Ghizlane, B. Hicham, and F. H. Reda, "A New Model of Automatic and Continuous Online Exam Monitoring," in *Proceedings - 2019 4th International Conference on Systems of Collaboration, Big Data, Internet of Things and Security, SysCoBIoTS 2019*, 2019, pp. 1–5, doi: 10.1109/SysCoBIoTS48768.2019.9028027.

[4] O. Holden, V. Kuhlmeier, and M. Norris, "Academic Integrity in Online Testing: A Research Review," 2020, doi: 10.31234/osf.io/rjk7g.

[5] C. Y. Chuang, S. D. Craig, and J. Femiani, "Detecting probable cheating during online assessments based on time delay and head pose," *High. Educ. Res. Dev.*, vol. 36, no. 6, pp. 1123–1137, 2017, doi: 10.1080/07294360.2017.1303456.

[6] S. Hu, X. Jia, and Y. Fu, "Research on Abnormal Behavior Detection of Online Examination Based on Image Information," in *Proceedings - 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2018*, 2018, vol. 2, pp. 88–91, doi: 10.1109/IHMSC.2018.10127.

[7] R. Bawarith, D. Abdullah, D. Anas, and P. Dr., "E-exam Cheating Detection System," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 4, pp. 176–181, 2017, doi: 10.14569/ijacsa.2017.080425.

[8] P. K. Mungai and R. Huang, "Using keystroke dynamics in a multi-level architecture to protect online examinations from impersonation," in *2017 IEEE 2nd International Conference on Big Data Analysis, ICBDA 2017*, 2017, pp. 622–627, doi: 10.1109/ICBDA.2017.8078710.

[9] Ananya and S. Singh, "Keystroke Dynamics for Continuous Authentication," in *Proceedings of the 8th International Conference Confluence 2018 on Cloud Computing, Data Science and Engineering, Confluence 2018*, 2018, pp. 205–208, doi: 10.1109/CONFLUENCE.2018.8442703.

[10] S. Prathish, S. Athi Narayanan, and K. Bijlani, "An intelligent system for online exam monitoring," in *Proceedings - 2016 International Conference on Information Science, ICIS 2016*, 2017, pp. 138–143, doi: 10.1109/INFOSCI.2016.7845315.

[11] S. S. Ketab, N. L. Clarke, and P. S. Dowland, "A Robust e-Invigilation System Employing Multimodal Biometric Authentication," *Int. J. Inf. Educ. Technol.*, vol. 7, no. 11, pp. 796–802, 2017, doi: 10.18178/ijiet.2017.7.11.975.

[12] N. A. Mahadi, M. A. Mohamed, A. I. Mohamad, M. Makhtar, M. F. A. Kadir, and M. Mamat, "A Survey of Machine Learning Techniques for Behavioral-Based Biometric User Authentication," *Recent Adv. Cryptogr. Netw. Secur.*, pp. 43–54, 2018, doi: 10.5772/intechopen.76685.

[13] A. Shdaifat, R. Obeidallah, G. Ghazal, A. A. Srhan, and N. R. Abu Spetan, "A proposed iris recognition model for authentication in mobile exams," *Int. J. Emerg. Technol. Learn.*, vol. 15, no. 12, pp. 205–216, 2020, doi: 10.3991/ijet.v15i12.13741.

[14] K. Garg, K. Verma, K. Patidar, N. Tejra, and K. Petidar, "Convolutional Neural Network based Virtual Exam Controller," in *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, 2020, pp. 895–899, doi: 10.1109/ICICCS48265.2020.9120966.

[15] N. I. Vivian and O. Anderson Ise, "Face Recognition Service Model for Student Identity Verification Using Deep Neural Network and Support Vector Machine (SVM)," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, pp. 11–20, 2020, doi: 10.32628/cseit2063225.

[16] M. Cote, F. Jean, A. B. Albu, and D. Capson, "Video summarization for remote invigilation of online exams," in *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, 2016, pp. 1–9, doi: 10.1109/WACV.2016.7477704.

[17] K. Jalali and F. Noorbehbahani, "An Automatic Method for Cheating Detection in Online Exams by Processing the Students Webcam Images," in *3rd Conference on Electrical and Computer Engineering Technology (E-Tech 2017), Tehran, Iran*, 2017, no. March.

[18] C. A, D. D, M. N, and M. B S, "a Survey on Detection of Anomalous Behaviour in Examination Hall," *Int. J. Eng. Appl. Sci. Technol.*, vol. 5, no. 2, pp. 583–588, 2020, doi: 10.33564/ijeast.2020.v05i02.098.

[19] J. Golden and M. Kohlbeck, "Addressing cheating when using test bank questions in online Classes," *J. Account. Educ.*, vol. 52, p. 100671, 2020, doi: 10.1016/j.jaccedu.2020.100671.

[20] V. A. Sangalli, G. Martinez-Munoz, and E. P. Canabate, "Identifying cheating users in online courses," in *IEEE Global Engineering Education Conference, EDUCON*, 2020, vol. 2020-April, pp. 1168–1175, doi: 10.1109/EDUCON45650.2020.9125252.

[21] S. Kausar, X. Huahu, A. Ullah, Z. Wenhao, and M. Y. Shabir, "Fog-Assisted Secure Data Exchange for Examination and Testing in E-learning System," *Mob. Networks Appl.*, pp. 1–17, 2020, doi: 10.1007/s11036-019-01429-x.

[22] "https://moodle.org/."

[23] A. W. Qurashi, V. Holmes, and A. P. Johnson, "Document Processing: Methods for Semantic Text Similarity Analysis," in *INISTA 2020 - 2020 International Conference on INnovations in Intelligent SysTems and Applications, Proceedings*, 2020, pp. 1–6, doi: 10.1109/INISTA49547.2020.9194665.

[24] M. Afzali and S. Kumar, "Comparative Analysis of Various Similarity Measures for Finding Similarity of Two Documents," *Int. J. Database Theory Appl.*, vol. 10, no. 2, pp. 23–30, 2017, doi: 10.14257/ijdta.2017.10.2.02.

[25] W. H.Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 13–18, 2013, doi: 10.5120/11638-7118.

[26] V. M.K and K. K, “A Survey on Similarity Measures in Text Mining,” *Mach. Learn. Appl. An Int. J.*, vol. 3, no. 1, pp. 19–28, 2016, doi: 10.5121/mlaij.2016.3103.

[27] G. Jain, T. Mahara, and K. N. Tripathi, “A Survey of Similarity Measures for Collaborative Filtering-Based Recommender System,” in *Advances in Intelligent Systems and Computing*, vol. 1053, Springer, 2020, pp. 343–352.

[28] K. P. Reddy, T. R. Reddy, G. A. Naidu, and B. Vishnu, “Impact of Similarity Measures in Information Retrieval,” pp. 54–59, 2018.

[29] M. Afzali and S. Kumar, “An Extensive Study of Similarity and Dissimilarity Measures Used for Text Document Clustering using K-means Algorithm,” *IJ Inf. Technol. Comput. Sci*, vol. 9, pp. 64–73, 2018.

[30] S. K. Pandey, B. Mishra, and S. S. Gautam, “Cluster Based Mining for Prediction of Heart Disease,” 2020.

[31] R. RAMAKRISHNAN, “A SURVEY ON STUDENTS PLACEMENT PERFORMANCE ANALYSIS USING WEKA TOOL.”

[32] P. Singh and A. Surya, “Performance analysis of clustering algorithms in data mining in weka,” *Int. J. Adv. Eng. Technol.*, vol. 7, no. 6, p. 1866, 2015.

[33] S. Gnanapriya, “Evaluation of Clustering Capability Using Weka Tool,” *Int. J. Innov. Eng. Technol.*, vol. 8, no. 1, pp. 181–187, 2017, doi: 10.21172/ijiet.81.025.

[34] H. Kaur and P. Verma, “Comparative Weka Analysis of Clustering Algorithm‘s,” *Int. J. Inf. Technol. Comput. Sci.*, vol. 9, no. 8, pp. 56–67, 2017, doi: 10.5815/ijitcs.2017.08.07.

[35] G. Sehgal and D. Garg, “Comparison of Various Clustering Algorithms,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 3, pp. 3074–307, 2014.

[36] J. A. Adeyiga, S. O. Olabiyisi, and E. O. Omidiora, “A comparative analysis of selected clustering algorithms for criminal profiling,” *Niger. J. Technol.*, vol. 39, no. 2, pp. 464–471, 2020, doi: 10.4314/njt.v39i2.16.