INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Cheating Detection for Online Examination Using Clustering Based Approach

Seng Zi Ong [a,*], Tee Connie [a], Michael Kah Ong Goh [a]

[a] *FIST, Multimedia University, Jalan Ayer Keroh Lama, 75450, Melaka, Malaysia*

Corresponding author: [*]1181203438@student.mmu.edu.my

***Abstract*— Online exams have become increasingly popular due to their convenience in eliminating the need for physical exams and allowing students to take exams from remote locations. However, one of the drawbacks of online exams is that they make cheating easier, and it can be difficult for online proctoring to detect subtle movements by the students. This could lead to doubts about students' exam results' value and overall credibility. To address this pressing issue, we present a cheating detection method using a CCTV camera to monitor students' faces, eyes, and devices to determine whether they cheat during exams. If suspicious behavior indicative of cheating is detected, a warning is raised to alert the students. A custom dataset was developed to train the model. The dataset consisted of recordings of pre-determined cheating behavior by 50 participants. These videos captured various poses and behaviors encoded and analyzed using a clustering approach. The encoded clustering method continuously tracks the students' faces, eyes, and body gestures throughout an exam. Experimental results show that the proposed approach effectively detects cheating behavior with a favorable accuracy of 83%. The proposed method offers a promising solution to the growing concern about cheating in online exams. This approach can significantly enhance the integrity and reliability of online assessment processes, fostering trust among educational institutions and stakeholders.**

***Keywords*—Cheating detection; online examination; object detection; clustering; machine learning.**

Manuscript received 4 Jan. 2023; revised 11 Apr. 2023; accepted 22 Aug. 2023. Date of publication 30 Nov. 2023.

I. INTRODUCTION

During 2019, COVID-19 began to spread and wreak havoc around the world. In response, educational institutions swiftly transitioned to online classes and exams [1]–[3]. While this approach resolved many challenges caused by COVID-19, it also introduced several advantages associated with online accessibility. Online exams allowed students to take tests from any location with Wi-Fi, eliminating the need for them to travel to specific exam centers [4], [5]. Furthermore, invigilators found it more convenient to monitor students by combining camera videos to observe their behavior [6]–[8].

However, the convenience of online exams also introduced a significant drawback: an increased potential for cheating [9]. Students could exploit the lack of physical supervision and attempt to cheat by seeking answers or engaging in dishonest practices off-camera without the teacher's detection [10]–[12]. Conversely, online proctoring is time-consuming and labor-intensive [13]. This is because online proctoring cannot monitor all the students simultaneously, and there are bound to be students who cheat when the teacher is not looking. There are many ways to cheat, and the system is too late to update those new cheating behaviors [2].

Therefore, this study aims to propose a cost-effective approach for detecting cheating behavior in online exams using machine learning approaches [14]–[16]. By utilizing a simple web camera, the study tracks and monitors the visual behavior of students, including facial expressions, body gestures, and eye movements, employing a robust clustering-based object detection method. For the study, fifty students from Multimedia University are invited to participate. These students take online exams while their behaviors are recorded, following specific instructions provided. The following sections provide a literature review of the existing methods in cheating detection.

A. Conventional Methods

In 2013, Javed and Aslam [17] developed a human, face, and eye detection method using an age detection and Kalman filtration algorithm. Their proposed system could detect eye movement and pupil behavior to determine if a student was cheating. However, this system had a weakness because it

lacked object detection capabilities to identify smartphones, notes, and other cheating devices. It also did not incorporate voice analysis to detect the sound of someone else providing answers to the examiner.

In 2017, Atoum et al. [10] used a multimedia analytics system for online exam proctoring. They employed audio-visual observation as their modality and utilized an SVM Classifier. The system demonstrated a high segment-based detection rate, which was a strength. However, a drawback of this system was the requirement for two cameras to detect cheating behavior effectively.

In the same year, Bawarith et al. [18] investigated and addressed various cheating methods in online exams. Their approach involved eye detection as the modality and utilized an equation-based method. The system could detect eye movement and pupil behavior to determine if a student was cheating. However, similar to the previous systems, it lacked object detection capabilities and voice analysis to detect external assistance or sounds.

On the other hand, Ghizlane et al. [19] proposed an Online Exam Management system to prevent cheating using machine learning algorithms. They employed face detection as the modality and used learning rules. The system utilized facial expressions to detect if an examiner was cheating. However, it did not include features for checking the student's browser activity or voice analysis.

In addition, Özgen et al. [20] developed an online interview anti-cheating system. Their approach involved object and face detection using a HOG-based SVM detector. The system did not require two cameras and could detect cheating behavior. However, it lacked features such as voice analysis, browser detection, and recognizing cheating without relying solely on facial expressions. Table I provides a summary of the conventional methods proposed for cheating detection.

TABLE I
RESULT OF DATA COLLECTION

Sources	Modal	Method	Dataset	Recognition Rate	Pros	Cons
[3]	Human Detection, Face Detection, Eye Detection	Age detection plus Kalman filtration algorithm	Dataset name: N/A Sample Size: 10 videos	93%	Can detect the eye movement and pupil to detect student is cheating or not	Does not have objection detection to detect the smartphone, note, and so on, and voice analysis to detect the sound of another person who is answering the examiner
[4]	Audio-visual observation	SVM Classifier	Dataset name: OEP Sample Size: 24	87%	Having a high segment-based detection rate	Need two cameras to conduct the anti-cheating
[5]	Eye Detection	Equation	Dataset name: N/A Sample Size: 30	97.78%	Can detect the eye movement and pupil to detect student is cheating or not	Does not have objection detection to detect the smartphone, note, and so on, and voice analysis to detect the sound of another person who is answering the examiner
[6]	Face detection	Learning rules	Dataset name: N/A Sample Size: N/A	N/A	Using facial expression to detect examiner is cheating or not	Do not check the student browser and voice analysis
[7]	Object Detection, Face detection	HOG based SVM detector	Dataset name: N/A Sample Size: 43	88%	No need two cameras and can detect	Does not have voice analyzing, browser detection and Recognize cheating without facial expressions.

B. Deep Learning Methods

Tiong and Lee [21] employed network IP detection and deep learning-based behavior detection techniques and utilized DenseLSTM to mitigate cheating in online exams. The system's strength lies in its ability to identify cheating by analyzing the speed at which students answer questions without the need for cameras to monitor students' faces. However, a weakness of this system is the absence of facial monitoring to observe potential cheating behavior directly.

On the other hand, Jadi [22] utilized facial expression detection and browser detection, employing CNN as the method to detect cheating. This system only required one camera and could detect facial expressions while preventing the opening of other applications. However, it did not incorporate voice analysis, which could be considered a limitation.

Dilini et al. [23] developed a browser extension that utilized eye tracking, employing face detection and OCSVM, to identify cheating behavior. The system's strength lies in its compatibility with web browser-based testing platforms, enhancing the quality of exams. However, it may be unable to detect cheating if the examiner uses notes affixed to the screen.

In addition, Barrientos et al. [24] aimed to leverage Amazon Web Services (AWS) to detect dishonest behavior among examiners, such as using smartphones for searching answers or detecting plagiarism. They employed face detection and TensorFlow as the method. The system's strength lies in its utilization of open-source sources and its

ability to recognize human faces and objects. However, it requires the purchase of AWS services.

Soltane and Laouar [25] developed an intelligent detection and recognition system using face and sound detection, employing CNN. The system's strength lies in using only one camera and microphone. However, it does not include a check of the examiner's browser activity. A summary of the deep learning approaches for cheating detection is presented in Table II.

TABLE II
DEEP LEARNING METHODS

Sources	Modal	Method	Dataset	Recognition Rate	Pros	Cons
[8]	Browser detection	DenseLSTM	Dataset name: 7wiseup Sample Size: 94	95.32%	No need camera to detect the speed at which students answer questions to identify any cheating	No cameras to monitor students' faces to see if they are cheating
[9]	Face detection, browser detection	CNN	Dataset name: N/A Sample Size: 10	97%	Just one camera that can detect facial expressions and not open other applications.	Does not have voice analysis.
[10]	Face detection	OCSVM	Dataset name: WebGazer Sample Size: 15000	92.04%	any web browser-based testing platform can utilize a plug-in to improve the quality of the exam	if the examiner uses notes glued to the screen, then this cheating cannot be detected
[11]	Face Detection	TensorFlow	Dataset name: N/A Sample Size: 30	87%	Most source is open-source and can recognize human face and object.	Need to purchase AWS system
[25]	Face detection, Sound Detection	CNN	Dataset name: LFW Sample Size: N/A	99.38%	Using one camera and one microphone to	This system does not check with the examiner's browser.

II. Material and Method

A web camera is utilized to monitor student behavior during an online exam. The system captures approximately 50 video frames at a time for subsequent analysis. The face, body gestures then process these frames and eye tracking model to determine the face and eyes' position and identify suspicious objects, such as a mobile phone. Once the positions are detected, the results are forwarded to the cheating detection module. This module examines the student's movements to identify potential cheating behavior. If the system detects a student using a mobile phone to cheat or observing answers by turning their head, an alert message is promptly sent to the student. However, if no cheating is detected, the system continues to monitor the student's behavior until the conclusion of the exam. The block diagram of the proposed method is illustrated in Fig. 1.

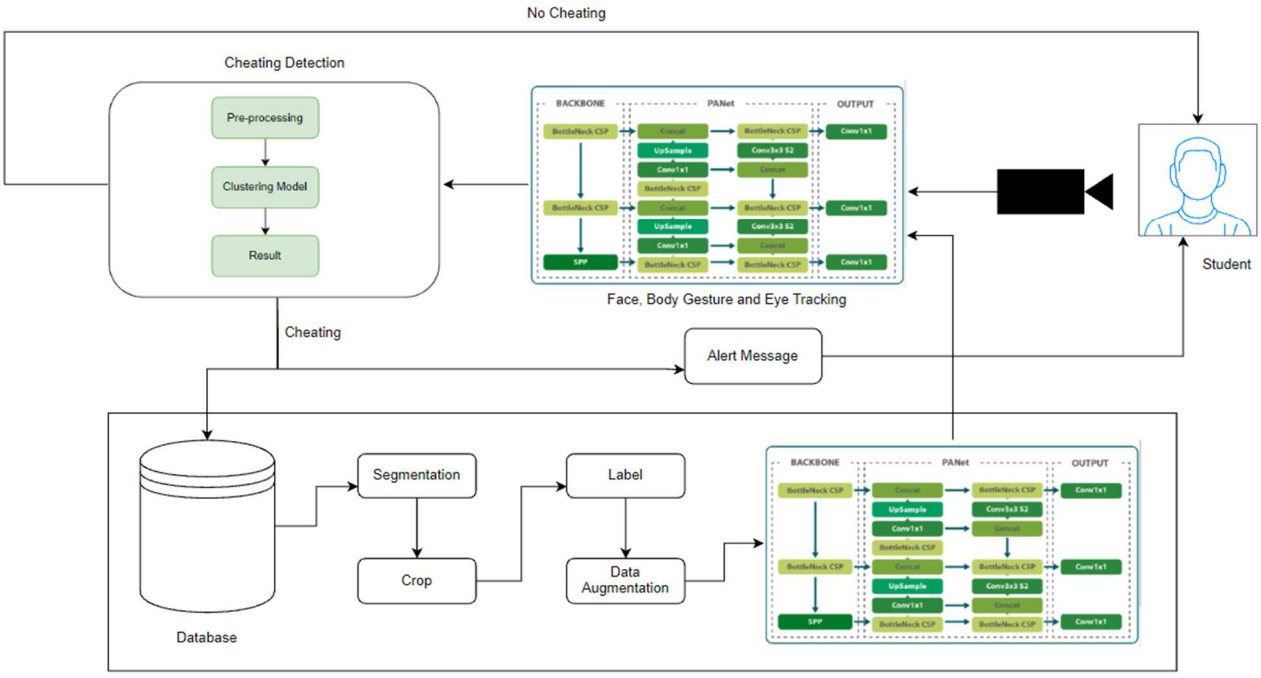

Fig. 1 Cluster Table of Time Series K-Means

A. Data Collection

In this study, four scenarios have been identified as cheating behavior: holding a smartphone, head movement, eye movement, and blocking the eyes with the hand. These scenarios are carefully selected to ensure that participants adhere to the prescribed steps and requirements, thus maintaining the relevance and integrity of the collected data. To facilitate data collection, Multimedia University, Melaka campus students have been invited to participate in this study. Before their involvement, each participant was given a consent form to ensure their willingness to contribute to the research.

In order to streamline the process of video recording, a Google session link was created and shared with the participants. This allowed for the utilization of recording software, specifically OBS Studio, which captured the participants' facial expressions exclusively. The video resolution was set to 1920 x 1080, and prior to recording, participants were shown a demonstration video to familiarize themselves with the process.

Data collection continued until 50 videos were obtained, which were then used for training. Each video was named according to the following format: 's_numPerson_scenario.mp4'. For instance, the third scenario involving the second person would be denoted as 's_2_3.mp4'. In total, we collected a dataset comprising 200 samples. Detailed participant characteristics and demographic information are summarized in Table III.

TABLE III
RESULT OF DATA COLLECTION

Number of Scenarios	4
Number of Participants	50
Number of Samples	200
Range of Age	19 - 67
Number of Males	39
Number of Females	11
Video Resolution	1920 x 1080
Recording Software	OBS Studio
Platform	Google Meet

B. Pre-processing

1) Segmentation: To facilitate the analysis process, Filmora's video editing software was employed for video segmentation purposes. In the first scenario, the video segments were categorized based on the placement of the phone, including phone center, phone left, and phone right. As a result, the video was split into three segments, and the corresponding location name was appended to each video name. For example, the video capturing the person looking at the phone on the right was named "s_1_1_right".In the second scenario, which involved various head movements, the video was further divided into eight segments: face down, face left, face right, face up, face down peeking, face left peeking, face right peeking, and face up peeking. Each segment was assigned a position name to accurately represent the movements depicted. These position names were added to the respective video names, ensuring clarity and categorization within the dataset. Similarly, the third scenario focused on eye movements and was divided into eight segments: eyes down, eyes left, eyes right, eyes up, eyes down peek, eyes left peek, eyes right peek, and eyes up peek. By splitting the videos and incorporating the position names in the video names, the dataset was organized and made ready for analysis. Finally, in the fourth scenario where only the eyes were blocked, no further segmentation was required as the focus remained solely on this aspect. In order to effectively distinguish between each scenario and behavior, a consistent naming convention was adopted. The following naming convention was utilized: look_down, look_up, look_side, look_center, eye_close, face_down, face_side, face_right, face_center, phone, and block_eye. The results obtained from the video analysis are presented in a structured format. Reading from left to right, the result format includes the following information: 'class', 'x_center', 'y_center', 'width', and 'height'. This format provides essential details about the identified objects or regions of interest within each frame of the videos.

2) Data Augmentation: Training and validating deep neural networks typically require substantial data, ideally in the thousands. However, with only 30 data points per category, the available data is clearly insufficient for effective training. To address this challenge, the "Albumentations" library was employed as a solution. Albumentations offers a powerful and user-friendly interface for image augmentation in various computer vision applications, including deep learning studies, object classification, segmentation, and detection. Utilizing Albumentations, a wide range of image transformation techniques can be applied to enhance the dataset and optimize model performance. These techniques include random cropping, flipping, and luminance contrast adjustment. One notable feature of Albumentations is its ability to handle bounding box parameters when working with annotated images. During the image enhancement process, the library recalculates the bounding boxes based on the modified images, ensuring the annotations remain accurate and unaffected. To increase the dataset size, the image enhancer was applied approximately five times to each image, resulting in a total of approximately 147,450 augmented data samples for training purposes. To perform validation, 2/5 of the augmented data were separated and saved in a designated repository named 'val', while the remaining data were stored in a repository named 'train'. Following the augmentation process, the enhanced images and corresponding new labels were organized and saved in a repository named 'aug_data'. This augmented dataset will serve as a valuable resource for training and validating the model.

C. Cheating Behavior Detection

Accurately detecting facial and eye positions plays a crucial role in determining a student's pose in front of the camera. For this purpose, YOLOv5, a well-known deep neural network, is utilized to locate landmark features. YOLOv5 belongs to the class of single-stage object detection models and offers several variants, including YOLOv5nm, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. In this study, YOLOv5s is chosen due to its optimal balance between accuracy and speed.

However, YOLOv5 alone is insufficient for training the data. To address this limitation, transfer learning is employed to incorporate the new and pre-existing YOLO models. Transfer learning enables the application of knowledge gained from training one machine learning model to another

related problem. In this case, the weights learned by the network in 'Task A' are transferred to a new 'Task B'. This approach allows for using learned knowledge to improve generalization and performance for a different task.

Once the system can successfully identify the positions of the face, eyes, and phone, the next step involves analyzing whether these positions indicate cheating behavior. Therefore, a cheating detection algorithm is implemented using clustering mechanism. This study compared three clustering algorithms to identify the most suitable approach for cheating detection.

To prepare the data for analysis, the video is converted into a dataset in CSV format. The video is split into 50 frames to enable accurate position detection. The Face, Body Gesture and Eye Tracking models are employed, and for each video frame, the models provide the corresponding position name. These position names are then converted into integer values for easier clustering and more efficient use of training data. The value 1 represents the 'down' position, 2 represents 'up', 3 represents 'side', 4 represents 'center', and 5 represents 'eye close'. The values assigned to 'phone' and 'block eye' are 4 and 11, respectively. Refer Table IV for a clear indication for the indexing for each position. These values are summed to distinguish different behaviors without conflicts, as shown in Table V.

TABLE IV
INDICES FOR EACH STANDALONE BEHAVIOR

Class Name	Value
Look down	1
Look up	2
Look side	3
Look center	4
Eye close	5
Face down	1
Face up	2
Face side	3
Face center	4
Phone	4

TABLE V
CATEGORIZATION AND LABELING OF THE CHEATING BEHAVIORS

Behavior	Face	Eye	Other	Category
No Face	0	0	0	0
Facing down	1	1	0	2
Facing left, Facing right	3	0	0	3
Facing up	2	2	0	4
Looking down	1	4	0	5
Looking up	2	4	0	6
Looking left, Looking right	3	4	0	7
Facing center	4	4	0	8
Looking at phone in the center	1	4	4	9
Looking at phone in the left and right	3	3	4	10
Blocking Eye	-	-	11	11

In the 'Facing left' and 'Facing right' categories, the eyes have a value of 0 instead of 3. This adjustment is made to avoid conflicts between the sum of the left and right faces (which would be 6) and the sum of the eyes. Therefore, the values are sorted accordingly. Additionally, after processing the fourth scene, the system creates two new scenes: ' Facing center' and 'no face'. These additional scenes are included to enhance the system's learning capability.

After creating the dataset, data clean-up is performed. During the data cleaning process, the system checks for noisy values in the 'block eyes', 'face down', 'face left', 'face right', 'face up', 'eyes down', 'eyes left', 'eyes right', 'eyes up', 'phone center', 'phone left', and 'phone right' columns. If any noisy values are found, they are replaced by the maximum values. Additionally, activities other than 'phone center', 'phone left', and 'phone right' are replaced with 0 for all other activities.

D. Classification of Cheating Behavior

The classification of cheating behavior is performed through a clustering mechanism. Clustering is chosen as the classification algorithm because it does not require manually defining labels for each cheating behavior. Given the large number of cheating behaviors, it would be impractical to add labels individually. The clustering algorithms group similar cheating behaviors without the need for explicit labels. Since clustering methods typically work with single features, various features such as faces, eyes, and others are combined into a single aggregated feature by summing up their values. This approach allows for a comprehensive representation of the cheating behavior. Three popular time series clustering algorithms have been investigated: Self-Organizing Maps (SOM), Fuzzy C-Means, and Time Series K-Means. By comparing their performance, the study aims to identify the most suitable algorithm for the task at hand.

1) Self-organizing maps (SOM): SOM [26], [27] is an unsupervised technique that is used for the visualization and analysis of high-dimensional datasets. Typically, SOM helps render high-dimensional datasets into low-dimensional ones, e.g., from 3D to 2D. A characteristic of SOM is that it does not require a target vector. Therefore, each node is connected to the input and the nodes are not linked to each other. Below is the equation of SOM.

$$D(i, j) = \|X - W(i, j)\| \quad (1)$$

X represents the input vector, which typically has the same dimensionality as the weight vectors in the SOM. $W(i, j)$ represents the weight vector associated with the neuron at position (i, j) on the SOM grid. The distance calculation can be performed by taking the square root of the sum of the squared differences between corresponding elements of the input vector and the weight vector.

$$D(i,j) = \sqrt{(\sum((X_k - X_k(i,j))^2))} \quad (2)$$

SOM is considered a technique for dimensionality reduction because it has the ability to build maps. It can be applied to cheating detection by treating each node of the SOM as an intermediate representation for clustering. It is important to note that each data point within the cheating dataset may possess different attributes and sizes. In this study,16 clusters are selected to correspond with the 16 distinct cheating behaviors in the dataset.

2) Fuzzy C-Means: Fuzzy C-Means [16], [28] divides the clustered data for each data point into a number of clusters based on their degree. The closer the data is to a particular cluster center, the more data there is near the center. Given that fuzzy c-means clustering produces better results than K-means clustering, the two can be compared. This is because the inputs to fuzzy clustering may belong to many clustering groups. The training time will increase as more iterations require more resources to compute in training. The dataset should have fewer outliers to improve the clustering results, as fuzzy c-means are sensitive to outliers and can significantly impact the clustering results. For each data point x_i and cluster centroid c_j, the membership value u_{ij} represents the degree to which x_i belongs to cluster c_j. The membership value is calculated using the following equation:

$$u_{ij} = 1 / \sum ((||x_i - c_j|| / ||x_i - c_k||)(2 / (m - 1))) \quad (3)$$

Once the membership values are calculated, the cluster centroids are updated using the following equation:

$$c_j = \sum ((u_{ij}\, m) * x_i) / \sum (u_{ij}\, m) \quad (4)$$

3) Time Series K-Means: One clustering method that can meet the requirements for clustering datasets is time series K-Means [29]. Using an unsupervised data mining process known as time series clustering, data points can be grouped according to their similarity. The goals are to maximize data similarity within clusters and reduce the similarity between clusters. Furthermore, in order to improve the efficiency and accuracy of time series clustering, the Dynamic Time Warping (DTW) metric [30], [31] must be used. This is because DTW is one of the methods for determining the closeness of two-time series of different lengths and speeds. In addition, the square root of the sum of the squared distances between each element in X and its nearest point in Y is used to calculate DTW. The calculation of DTW distance involves constructing a matrix and finding the minimum cost path. The equation for DTW distance between two time series X_i and X_j is as follows:

$$dist(X_i, X_j) = \sqrt{(DTW(X_i, X_j))} \quad (5)$$

III. Result and Discussion

A. Evaluation of Face, Eye and Suspicious Objects Detection

In this study, all the images are resized to 450 x 450 pixels. Two different batch sizes have been experimented with 16 and 32. The batch size is adjusted based on the GPU's performance and the dataset's size. Tables VI and VII present the training data results using the different configurations of parameters. After analyzing the results, it was found that using 15 epochs provided the best performance. This decision is based on the balanced precision, recall, and the percentage of mAP_0.5 in the results. Moreover, the difference in performance between the two batch sizes was negligible. Therefore, the smallest batch size, which is 16, is selected for the subsequent tests.

TABLE VI
CATEGORIZATION AND LABELLING OF THE CHEATING BEHAVIORS

Batch	Epochs	Precision	Recall	mAP_0.5
16	8	91.29%	69.28%	78.64%
16	9	91.57%	69.96%	79.43%
16	10	91.92%	70.34%	80.06%
16	11	86.78%	71.03%	80.57%
16	12	89.32%	72.26%	81.05%
16	13	89.87%	72.93%	81.48%
16	14	90.02%	73.12%	81.89%
16	15	89.56%	73.62%	82.32%

TABLE VII
CATEGORIZATION AND LABELLING OF THE CHEATING BEHAVIOURS

Batch	Epochs	Precision	Recall	mAP_0.5
32	8	91.29%	69.28%	78.64%
32	9	91.57%	69.96%	79.43%
32	10	91.92%	70.34%	80.06%
32	11	86.78%	71.03%	80.57%
32	12	89.32%	72.26%	81.05%
32	13	89.87%	72.93%	81.48%
32	14	90.02%	73.12%	81.89%
32	15	89.56%	73.62%	82.32%

The detection results for the different scenarios are illustrated in Fig. 2 to Fig. 7. Based on these results, the proposed method demonstrates the ability to accurately identify the position of the face, eyes, and phone in different situations. For example, in Scenario 1 (Fig. 2), where the student is holding the phone and facing it directly, the method successfully detects the phone, face, and eye position. Scenario 2 (Fig. 3) illustrates a scenario with facial and eye movement. The method detects the upward-facing movement with the face looking up. In Scenario 3 (Fig. 4), the focus is on the eye movement without facial movement. The method accurately detects the downward gaze with the face-centered and the eyes looking upward. Scenario 4 (Fig. 5) demonstrates the detection of eye blocking, where the student uses their hand to cover their eye. The method effectively detects blocked eyes. Fig. 6 represents a no-cheating scenario where the student faces and looks toward the center. In this case, the method does not raise alerts or detect any specific cheating behavior. The last figure (Fig. 7) indicates a situation where the camera fails to detect anything, particularly the student's face. In such cases, the computer generates an alert message to adjust the camera. Overall, the detection results showcased in these figures demonstrate the method's ability to identify various cheating behaviors reliably.

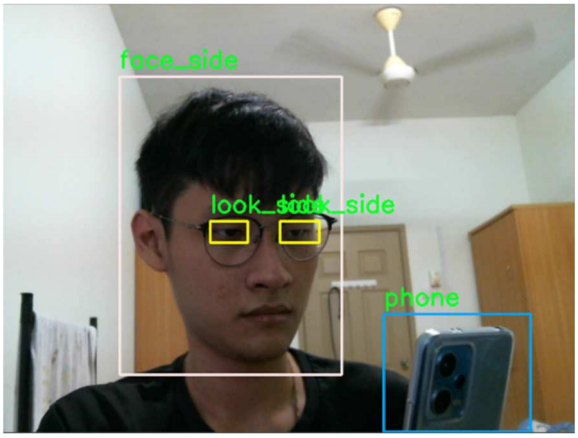

Fig. 2 Scenario 1

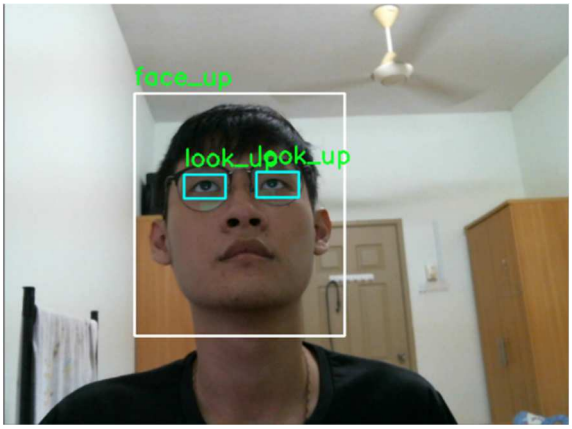

Fig. 3 Scenario 1

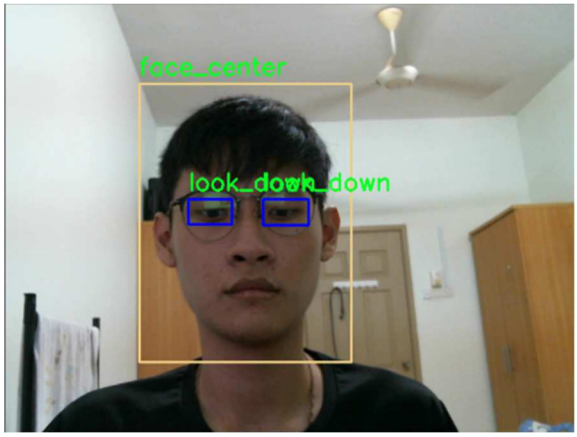

Fig. 4 Scenario 3

Fig. 5 Scenario 4

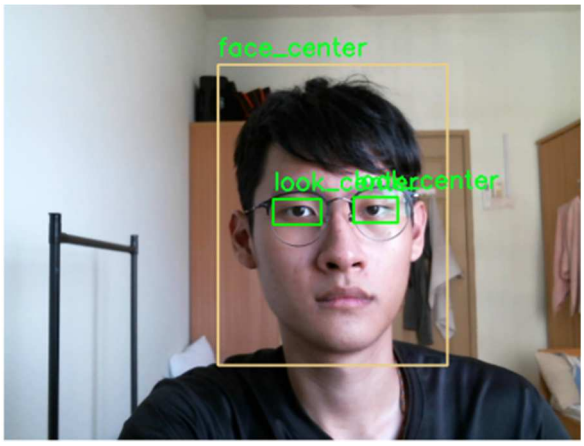

Fig. 6 Scenario 3

Fig. 7 Scenario 4

B. Results of Cheating Detection

This section evaluates the performance of the proposed cheating detection approach. A table resembling a confusion matrix is presented to determine which clusters most cheating behaviors belong to. To calculate the accuracy of each cluster, the maximum value within the cluster is considered for each cheating behavior, and this value is divided by the total number of occurrences of that cheating behavior. The cluster with the highest maximum value is chosen as the final cluster for that specific cheat. The mean function is also utilized to calculate the average accuracy across all clusters.

Table VIII displays the results obtained from the SOM, Fuzzy C-Means, and time series K-Means algorithms. The Accuracy is the average accuracy of the clustering algorithm and cluster's accuracy less than 50% accuracy is the number of each cluster's accuracy which is less than 50%. Among these methods, time series K-Means exhibited the highest accuracy. Furthermore, no cluster's accuracy was less than 50% in the time series K-Means algorithm. Fig. 8 presents the cluster table specifically for time series K-Means. Please note that the presented evaluation results demonstrate the effectiveness of the proposed cheating detection approach based on the applied algorithms. However, it is important to consider other factors, such as dataset characteristics and specific requirements of the application, when selecting the most suitable clustering algorithm for a given scenario.

	Cluster 0	Cluster 1	Cluster 10	Cluster 11	Cluster 12	Cluster 13	Cluster 14	Cluster 15	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9	Accuracy	True Cluster
block eye	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	1.0	0.0	47.0	92.16	Cluster 9
eye down	0.0	1.0	0.0	0.0	45.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	91.84	Cluster 12
eye left	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	48.0	0.0	0.0	97.96	Cluster 7
eye peek down	0.0	1.0	1.0	0.0	4.0	1.0	34.0	0.0	4.0	0.0	0.0	0.0	1.0	2.0	0.0	0.0	70.83	Cluster 14
eye peek left	0.0	0.0	0.0	1.0	0.0	0.0	11.0	0.0	1.0	0.0	0.0	0.0	0.0	26.0	0.0	0.0	66.67	Cluster 7
eye peek right	0.0	0.0	0.0	0.0	1.0	0.0	8.0	0.0	4.0	0.0	0.0	0.0	0.0	24.0	0.0	0.0	64.86	Cluster 7
eye peek up	0.0	3.0	2.0	1.0	2.0	0.0	7.0	0.0	31.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	64.58	Cluster 2
eye right	1.0	1.0	0.0	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	44.0	0.0	0.0	89.80	Cluster 7
eye up	3.0	5.0	0.0	40.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	81.63	Cluster 11
face center	0.0	69.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.00	Cluster 1
face down	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	1.0	0.0	44.0	0.0	0.0	0.0	0.0	88.00	Cluster 5
face left	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	49.0	0.0	98.00	Cluster 8
face peek down	0.0	0.0	8.0	0.0	0.0	3.0	2.0	0.0	0.0	0.0	0.0	0.0	36.0	0.0	0.0	0.0	73.47	Cluster 6
face peek left	1.0	0.0	0.0	0.0	0.0	48.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	97.96	Cluster 13
face peek right	0.0	0.0	0.0	0.0	0.0	49.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.00	Cluster 13
face peek up	0.0	1.0	34.0	0.0	1.0	0.0	11.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	69.39	Cluster 10
face right	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.0	0.0	100.00	Cluster 8
face up	48.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.00	Cluster 0
no face	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	100.00	Cluster 3
phone center	2.0	0.0	0.0	17.0	3.0	0.0	0.0	27.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	54.00	Cluster 15
phone left	0.0	1.0	0.0	6.0	0.0	0.0	0.0	1.0	0.0	0.0	32.0	1.0	0.0	0.0	0.0	9.0	64.00	Cluster 4
phone right	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	39.0	0.0	0.0	0.0	0.0	6.0	78.00	Cluster 4

Fig. 8 Cluster Table of Time Series K-Means

TABLE VIII
CATEGORIZATION AND LABELING OF THE CHEATING BEHAVIORS

Clustering Algorithm	Accuracy (%)	Cluster's accuracy < 50%
SOM	82.35	1
Fuzzy C-Means	77.97	4
Time Series K-Means	83.6	0

The accuracy of all clusters is above 50%, indicating that each cluster can predict the motion well. However, the cluster with the lowest accuracy is 'phone center'. Notably, 17 instances of 'phone center' were classified into cluster 11. This discrepancy could be attributed to noisy data within the 'phone center' dataset, resulting in incorrect allocation to the corresponding cluster. Interestingly, both 'phone left' and 'phone right' were assigned to the same cluster, namely cluster 4. This can be attributed to the fact that their category values, representing the left and right positions, are considered as part of the 'side' category. Similar situations were observed with 'face left' and 'face right', 'face peek left' and 'face peek right', 'eye left' and 'eye right', as well as 'eye look peek left' and 'eye peek right', all being clustered together.

Furthermore, 'eye left', 'look right', 'eye peek left', and 'eye peek right' were grouped into cluster 7. This could be because the motion patterns of 'eye peek left' and 'eye peek right' are not significantly different from those of 'eye left' and 'eye right', resulting in some data being classified as static motions, similar to 'eye left' and 'eye right'. Another possibility is that due to the limited number of clusters, they are combined together.

These observations highlight the need for further investigation and refinement of the clustering approach, particularly in cases where similar motion patterns are categorized into different clusters. Additionally, increasing the number of clusters might help to better distinguish between such subtle variations in motion.

To visualize the movement or form of each cluster, the sequences belonging to a specific cluster were plotted in translucent grey. Additionally, the average sequence of that cluster was calculated and represented in red. Based on the plotted graph in Fig. 9, it can be observed that six clusters exhibit dynamic motion, namely cluster 2, cluster 6, cluster 7, cluster 10, cluster 13, and cluster 14. On the other hand, the remaining ten clusters represent static motion.

Referring to the Cluster Table of Time Series K-Means, it can be determined that cluster 2 corresponds to 'eye peek up', cluster 6 corresponds to 'face peek down', cluster 7 corresponds to 'eye left', 'look right', 'eye peek left', and 'eye peek right', cluster 10 corresponds to 'face peek up', cluster 13 corresponds to 'face peek left' and 'face peek right', and cluster 14 corresponds to 'eye peek down'. Notably, cluster 7 includes dynamic and static motions such as 'eye left' and 'eye right', which can result in a relatively stationary graph. These visualizations provide insights into the distinct movement patterns captured by each cluster, highlighting the presence of both dynamic and static motion clusters and aiding in interpreting the clustering results.

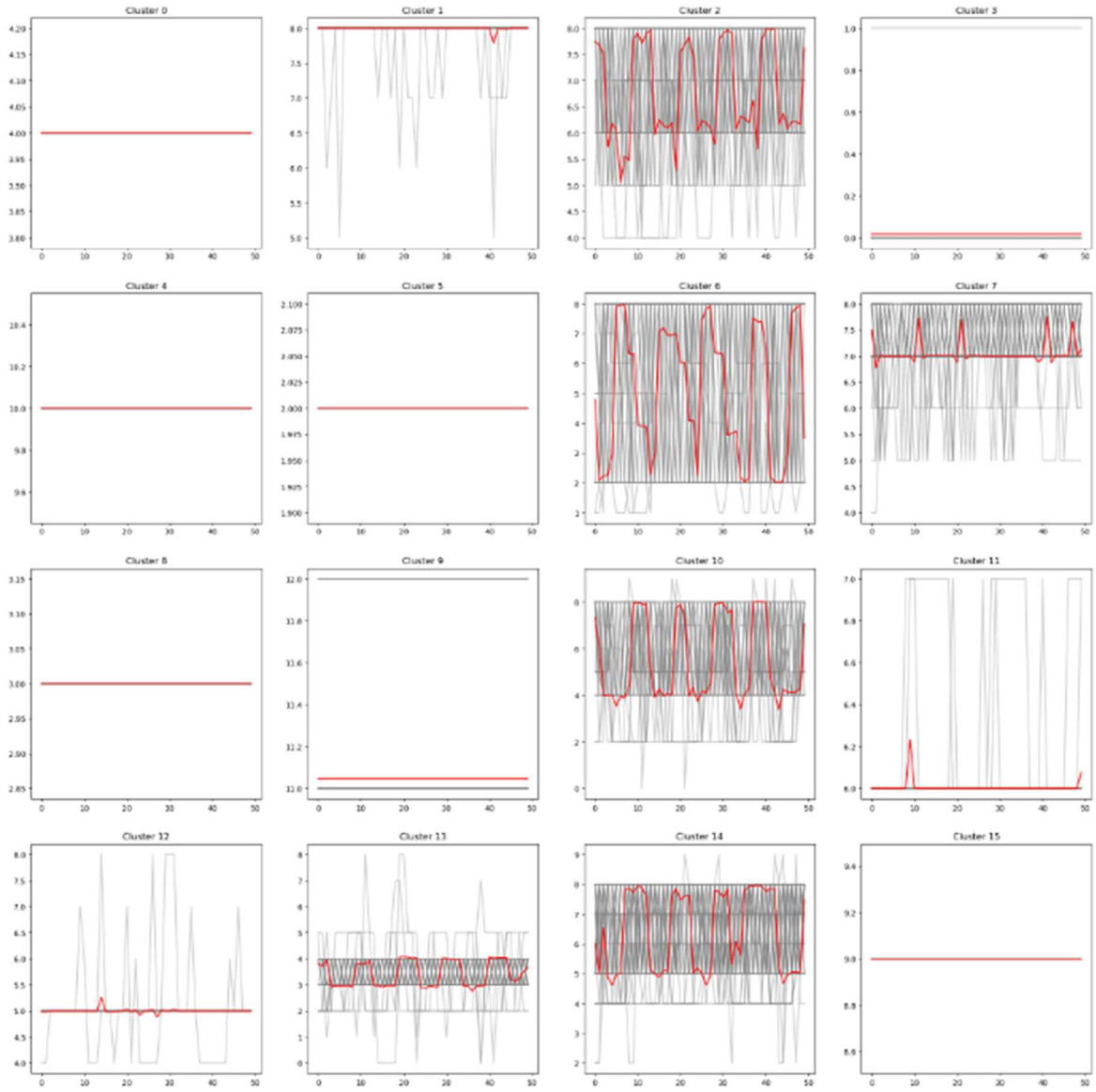

Fig. 9 Plot Result of Time Series K-Means.

Fig. 10 illustrates the cluster distribution resulting from the Time Series K-Means algorithm. Examining the clustering distribution, it becomes evident that all clusters have successfully grouped the data. The cluster with the highest representation is cluster 7, which can be attributed to its inclusion of four distinct behaviors: 'eye left', 'look right', 'eye peek left', and 'eye peek right'. Additionally, cluster 8 and cluster 13 have the second-highest values as they encompass two behaviors each.

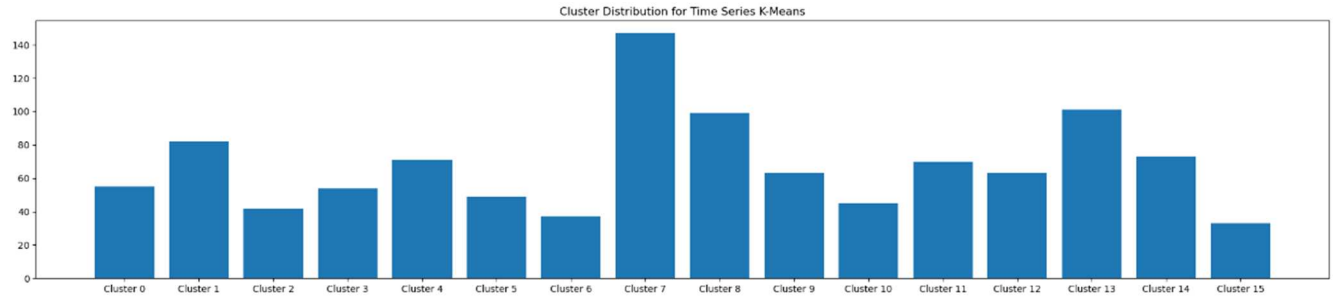

Fig. 10 Cluster Distribution of Time Series K-Means

Overall, the average accuracy achieved by the clustering algorithm is 83.60%. This indicates that all cheating behaviors have been correctly clustered, contributing to the effectiveness of the proposed approach in detecting and categorizing cheating instances.

C. Comparison with State of the Art

Comparing the results of our clustering method to state-of-the-art classification methods, I observed notable differences in accuracy. Our clustering method achieved an accuracy of 83%. In contrast, the classification methods presented in other articles achieved accuracies of 88%, 92.04%, and 95.32%, respectively. These findings indicate that the classification methods outperform our clustering approach in terms of accuracy.

However, it is important to note that clustering and classification are distinct techniques with different objectives. While classification aims to assign instances to predefined classes, clustering seeks to discover inherent patterns and group similar instances together without prior knowledge of class labels. By using classification methods, when the computer detects a new cheat, it will not be able to classify it because the new cheat is not in the training dataset. Consequently, clustering methods allow computers to cluster cheating behavior without the need for labels.

TABLE IX
RESULT OF CHEATING DETECTION PIPELINE FOR ONLINE INTERVIEWS AND EXAMS

Scenario	Precision	Recall	F1
Another person	55%	86%	67%
Device	100%	83%	91%
Absence	89%	89%	86%
Overall cheating	90%	86%	88%

TABLE X
RESULT OF CHEATING DETECTION IN BROWSER-BASED ONLINE EXAMS THROUGH EYE GAZE TRACKING

Recall	94.55%
Precision	89.66%
Accuracy	89.53%
F1 Score	92.04%

TABLE XI
RESULT OF DETECTION OF CHEATING AT ONLINE EXAMINATIONS USING DEEP LEARNING APPROACH

Network	Mid-term (%)	Final term (%)	Overall (%)
DNN	82.74	52.68	67.71
LSTM	94.49	89.29	91.89
RNN	87.20	85.02	86.11
DenseLSTM	97.77	92.86	95.32

D. Discussions

From the obtained results, it can be concluded that the Time Series K-Means algorithm is the most suitable for training time series datasets. This is due to its ability to cluster cheating behaviors with dynamic motion accurately. On the other hand, SOM and Fuzzy C-Means algorithms are more suitable for static datasets and may not perform as well with time series data. These algorithms are better equipped to cluster stationary behaviors such as face left, eyes right, and center of the phone. However, they struggle to effectively cluster dynamic motion behaviors like face peeking down and eyes peeking left, which require capturing temporal patterns. Fig. 11 shows a sample interface for the proposed method implemented for cheating detection.

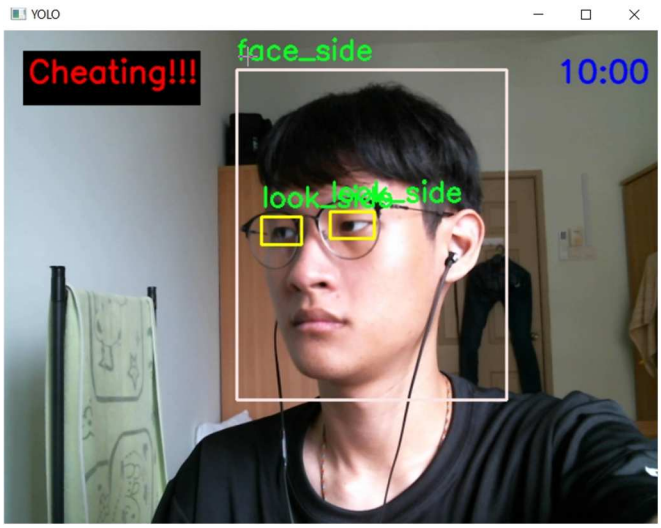

Fig. 11 Sample Interface for Cheating Detection System

IV. CONCLUSIONS

This study presents a robust cheating detection method for online examinations. The developed system effectively detects cheating behavior based on facial expressions, eye movements, and body posture with high accuracy. Additionally, the system saves a video recording of detected cheating behaviors in a designated folder. However, certain limitations have been identified. The system requires the camera to be positioned in the center. If the camera is placed at a different angle, the system may misinterpret the student's gaze direction, even when they are looking directly ahead. Currently, the system classifies "eye left" and "eye peek left" into the same cluster, potentially leading to misinterpretation.

To address these limitations, future improvements will be implemented. Additional data from various camera angles and scenarios involving multiple individuals will be collected. This expanded dataset will enable the system to learn and adapt to different camera placements and differentiate between multiple individuals' behaviors. Secondly, the pre-processing stage will be enhanced to improve the distinction between "eye left" and "eye peek left" behaviors. This could involve refining the clustering process and incorporating more clusters during training to achieve better separation.

ACKNOWLEDGMENT

This research is supported by the Fundamental Research Grant Scheme (FRGS/1/2020/ICT02/MMU/02/5).

REFERENCES

[1] U. Vellappan, L. Lim, and S. Y. Lim, "Engaging Learning Experience: Enhancing Productivity Software Lessons with Screencast Videos," *Journal of Informatics and Web Engineering*, vol. 2, no. 2, Art. no. 2, Sep. 2023, doi: 10.33093/jiwe.2023.2.2.14.

[2] I. N. Yulita, F. A. Hariz, I. Suryana, and A. S. Prabuwono, "Educational Innovation Faced with COVID-19: Deep Learning for Online Exam Cheating Detection," *Education Sciences*, vol. 13, no. 2, Art. no. 2, Feb. 2023, doi: 10.3390/educsci13020194.

[3] D. M. Cretu and Y.-S. Ho, "The Impact of COVID-19 on Educational Research: A Bibliometric Analysis," *Sustainability*, vol. 15, no. 6, Art. no. 6, Jan. 2023, doi: 10.3390/su15065219.

[4] M. Labayen, R. Vea, J. Flórez, N. Aginako, and B. Sierra, "Online Student Authentication and Proctoring System Based on Multimodal Biometrics Technology," *IEEE Access*, vol. 9, pp. 72398–72411, 2021, doi: 10.1109/ACCESS.2021.3079375.

[5] R. Wuthisatian, "Student exam performance in different proctored environments: Evidence from an online economics course," *International Review of Economics Education*, vol. 35, p. 100196, Nov. 2020, doi: 10.1016/j.iree.2020.100196.

[6] 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.

[7] K. Butler-Henderson and J. Crawford, "A systematic review of online examinations: A pedagogical innovation for scalable authentication and integrity," *Computers & Education*, vol. 159, p. 104024, Dec. 2020, doi: 10.1016/j.compedu.2020.104024.

[8] S. M. Aslam, A. K. Jilani, J. Sultana, and L. Almutairi, "Feature Evaluation of Emerging E-Learning Systems Using Machine Learning: An Extensive Survey," *IEEE Access*, vol. 9, pp. 69573–69587, 2021, doi: 10.1109/ACCESS.2021.3077663.

[9] S. Dendir and R. S. Maxwell, "Cheating in online courses: Evidence from online proctoring," *Computers in Human Behavior Reports*, vol. 2, p. 100033, Aug. 2020, doi: 10.1016/j.chbr.2020.100033.

[10] Y. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu, and X. Liu, "Automated Online Exam Proctoring," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1609–1624, Jul. 2017, doi: 10.1109/TMM.2017.2656064.

[11] M. E. Rodríguez, A.-E. Guerrero-Roldán, D. Baneres, and I. Noguera, "Students' Perceptions of and Behaviors Toward Cheating in Online Education," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 16, no. 2, pp. 134–142, May 2021, doi: 10.1109/RITA.2021.3089925.

[12] S. Mukherjee, B. Rohles, V. Distler, G. Lenzini, and V. Koenig, "The effects of privacy-non-invasive interventions on cheating prevention and user experience in unproctored online assessments: An empirical study," *Computers & Education*, vol. 207, p. 104925, Dec. 2023, doi: 10.1016/j.compedu.2023.104925.

[13] S. Kaddoura and A. Gumaei, "Towards effective and efficient online exam systems using deep learning-based cheating detection approach," *Intelligent Systems with Applications*, vol. 16, p. 200153, Nov. 2022, doi: 10.1016/j.iswa.2022.200153.

[14] R. Shafique, W. Aljedaani, F. Rustam, E. Lee, A. Mehmood, and G. S. Choi, "Role of Artificial Intelligence in Online Education: A Systematic Mapping Study," *IEEE Access*, vol. 11, pp. 52570–52584, 2023, doi: 10.1109/Access.2023.3278590.

[15] M. Garg and A. Goel, "Preserving integrity in online assessment using feature engineering and machine learning," *Expert Systems with Applications*, vol. 225, p. 120111, Sep. 2023, doi: 10.1016/j.eswa.2023.120111.

[16] E. F. Okagbue *et al.*, "A comprehensive overview of artificial intelligence and machine learning in education pedagogy: 21 Years (2000–2021) of research indexed in the scopus database," *Social Sciences & Humanities Open*, vol. 8, no. 1, p. 100655, Jan. 2023, doi: 10.1016/j.ssaho.2023.100655.

[17] A. Javed and Z. Aslam, "An Intelligent Alarm Based Visual Eye Tracking Algorithm for Cheating Free Examination System," *IJISA*, vol. 5, no. 10, pp. 86–92, Sep. 2013, doi: 10.5815/ijisa.2013.10.11.

[18] R. Bawarith, D. A. Basuhail, D. A. Fattouh, and P. D. S. Gamalel-Din, "E-exam Cheating Detection System," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 4, Art. no. 4, 53/29 2017, doi: 10.14569/IJACSA.2017.080425.

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

[20] A. C. Ozgen, M. U. Öztürk, O. Torun, J. Yang, and M. Z. Alparslan, "Cheating Detection Pipeline for Online Interviews," in *2021 29th Signal Processing and Communications Applications Conference (SIU)*, Jun. 2021, pp. 1–4. doi: 10.1109/SIU53274.2021.9477950.

[21] L. C. O. Tiong and H. J. Lee, "E-cheating Prevention Measures: Detection of Cheating at Online Examinations Using Deep Learning Approach -- A Case Study." arXiv, Jan. 24, 2021. doi: 10.48550/arXiv.2101.09841.

[22] A. Jadi, "New Detection Cheating Method of Online-Exams during COVID-19 Pandemic," *International Journal of Computer Science and Network Security*, vol. 21, no. 4, pp. 123–130, Apr. 2021, doi: 10.22937/IJCSNS.2021.21.4.17.

[23] N. Dilini, A. Senaratne, T. Yasarathna, N. Warnajith, and L. Seneviratne, "Cheating Detection in Browser-based Online Exams through Eye Gaze Tracking," in *2021 6th International Conference on Information Technology Research (ICITR)*, Dec. 2021, pp. 1–8. doi: 10.1109/ICITR54349.2021.9657277.

[24] A. Barrientos, M. Cuadros, J. Alba, and Á. S. Cruz, "Implementation of a remote system for the supervision of online exams through the use of cameras with artificial intelligence," in *2021 IEEE Engineering International Research Conference (EIRCON)*, Oct. 2021, pp. 1–4. doi: 10.1109/EIRCON52903.2021.9613352.

[25] M. Soltane and M. R. Laouar, "A Smart System to Detect Cheating in the Online Exam," in *2021 International Conference on Information Systems and Advanced Technologies (ICISAT)*, Dec. 2021, pp. 1–5. doi: 10.1109/ICISAT54145.2021.9678418.

[26] D. Steffen and A. Chaves Neto, "Ranking Model Applying Self-Organizing Maps and Factor Analysis," *IEEE Latin America Transactions*, vol. 19, no. 7, pp. 1217–1224, Jul. 2021, doi: 10.1109/TLA.2021.9461851.

[27] P. Yao, Q. Zhu, and R. Zhao, "Gaussian Mixture Model and Self-Organizing Map Neural-Network-Based Coverage for Target Search in Curve-Shape Area," IEEE Transactions on Cybernetics, vol. 52, no. 5, pp. 3971–3983, May 2022, doi: 10.1109/tcyb.2020.3019255.

[28] D. Kumar, R. K. Agrawal, and P. Kumar, "Bias-Corrected Intuitionistic Fuzzy C-Means With Spatial Neighborhood Information Approach for Human Brain MRI Image Segmentation," IEEE Transactions on Fuzzy Systems, vol. 30, no. 3, pp. 687–700, Mar. 2022, doi: 10.1109/tfuzz.2020.3044253.

[29] Z. Zhang et al., "Solar Radiation Intensity Probabilistic Forecasting Based on K-Means Time Series Clustering and Gaussian Process Regression," IEEE Access, vol. 9, pp. 89079–89092, 2021, doi: 10.1109/access.2021.3077475.

[30] L. Bai, L. Cui, Z. Zhang, L. Xu, Y. Wang, and E. R. Hancock, "Entropic Dynamic Time Warping Kernels for Co-Evolving Financial Time Series Analysis," IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 4, pp. 1808–1822, Apr. 2023, doi: 10.1109/tnnls.2020.3006738.

[31] T. Belkhouja, Y. Yan, and J. R. Doppa, "Dynamic Time Warping Based Adversarial Framework for Time-Series Domain," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 6, pp. 7353–7366, Jun. 2023, doi: 10.1109/tpami.2022.3224754.