Journal of Information Systems and Informatics

Vol. 7, No. 3, September 2025 e-ISSN: **2656-4882** p-ISSN: **2656-5935**

DOI: 10.51519/journalisi.v7i3.1262 Published By DRPM-UBD

Real-time Multi-Screen Cheating Detection using K-Means Clustering

Yudhi Setyo Purwanto[1], Rakhmadi Irfansyah Putra[2]

[1,2]Department of Informatics, Faculty of Energy Telematics, Institut Teknologi PLN
Email: [1]y.purwanto@itpln.ac.id, [2]rakhmadi@itpln.ac.id

Abstract

Ensuring academic honesty during online exams is becoming more and more challenging with students taking advantage of multiple screens and mirrored monitors. This research presents a privacy-sensitive, real-time multi-screen behavior detection model that does not rely on cameras or biometric sensors. The system tracks hardware and behavioral signals like screen-switch rate, focus-loss activity, idle time, and display change events. Utilizing K-Means clustering (k = 3), these metrics are segregated into three categories: Normal, Suspected, and Cheating. Implemented in Python and tested on simulated and real datasets, the model registered a silhouette score of 0.27 and showed discriminative behavior segregation through clustering analysis. Testing against a labelled dataset produced balanced accuracy of more than 80 percent, supported by confusion matrix and performance curve research. Findings show that hardware monitoring and activity-based could be an effective, camera-free means of detecting cheating in online examinations. The approach is privacy-respecting, computationally light in real time, and has understandable output for administrative exam. Drawbacks include the focus to date on Windows platforms and the need for more comprehensive cross-platform testing. Future studies will examine multimodal integration and larger scales to further increase detection accuracy and transferability.

Keywords: online exam, academic integrity, multi-screen cheating, K-Means clustering, behavioural analytics

1. INTRODUCTION

The rapid transition to online testing changed assessment practices but also fueled concern about academic integrity. Studies during the COVID-19 era identified prevalent issues in maintaining honesty in the realm of remote testing [1-2]. Although camera-based proctoring remains a common defense, it has the tendency to attract privacy grievances, needs stable internet connections, and cannot detect hardware-level cheating such as multiple screen or mirrored display usage [3-4]. Students can connect secondary monitors, utilize wireless displays, or use companion devices without triggering alarms, exposing the limitations of current visual and rule-based systems.

Other than technical loopholes, perceptions of fairness and oversight also influence cheating behavior. Van De Sande and Lu (2018) [5] and Valizadeh (2022) [6] disclose that students perceive online tests as more susceptible to manipulation and less proctored compared to in-person tests. Cotton et al. (2024) [7] also joined in that advances in generative AI created new forms of academic dishonesty, and it becomes more difficult to identify in online environments. These findings affirm the necessity of privacy-respecting, non-intrusive systems that detect cheating based on behavior and device usage rather than direct surveillance.

At the same time, researchers began to explore behavioral and machine learning approaches to proctoring monitoring. Heinrich (2025) [8] argued for open, interoperable proctoring standards to facilitate transparency and trust, while Peled et al. (2018) [9] and Taşkın and Kokoç (2025) [10] demonstrated how behavioral engagement patterns can signal dishonesty without requiring visual information. Henderson et al. (2024) [11] further highlighted that perceived fairness plays an important role in student acceptance of proctoring software.

At the technical level, research in clustering and data mining has shown promising outcomes for the detection of anomalous behavior in educational data. Oti et al. (2021) and Priyatma et al. (2024) [12-13] discussed and applied K-Means clustering as an effective unsupervised method of pattern detection in learning analytics, and Chen and Yu (2021) [14] demonstrated its use in online test data processing. Such models can classify data without prior labelling, a boon for identifying unexpected cheating behaviors.

Building on this precedent, the present study proposes a camera-free, real-time detection system that identifies multi-screen cheating behavior using unsupervised learning. The system logs hardware and behavioral metrics, such as screen-switching frequency, focus-loss events, idle time, and display configuration changes, and applies K-Means clustering (k = 3) to assign sessions to Normal, Suspected, and Cheating clusters. Unlike camera-based or rule-based systems, this model learns dynamically in varied user contexts without intruding on privacy or sacrificing interpretability.

The aim of this study is to create and test a lightweight, data-driven multi-screen cheating detection system for online examinations without infringing on user privacy. The following questions guide the study: 1) Can hardware and behavioral metrics efficiently separate normal and suspicious user behavior in online testing environments? 2) How effective is an unsupervised clustering model at detecting cheating patterns in real-time? And 3) What are the operational strategies that can lead to fair and transparent results for "Suspected" cases?

The novelty of the research lies in the fusion of behavioral analytics with device-level monitoring to create a camera-free, privacy-preserving detection system. It contributes to the growing trend of transparent and ethical digital examination in three main outputs: 1) Design of a real-time monitoring system that identifies secondary-screen connections and multitasking activities, 2) Deployment of a K-Means clustering model that automatically segments user sessions into interpretable behavioral classes, and 3) Empirical validation showing that hardware and behavioral signals can be utilized as reliable, non-invasive predictors of cheating behavior.

2. METHODOLOGY

The system to be proposed is based on a modular structure with three main layers: data collection, processing and clustering, and graphical output. The background monitoring module captures behavior and hardware information every five seconds, including focus loss, mouse idle, and changes to display settings. These streams are normalized and processed via the K-Means clustering algorithm, which maps behavior to Normal, Suspected, or Cheating clusters. The clustering results are then presented on an administrator dashboard and stored for post-exam evaluation.

This design aligns with previous plans of online test integrity, where unproctored testing environments require adaptive, data-driven solutions rather than camera observation (Sanz et al., 2020) [15]. When suspicious activity is logged, the system automatically freezes the participant's exam window and notifies the proctor through the central database. This process allows for real-time intervention alongside transparency in alert creation.

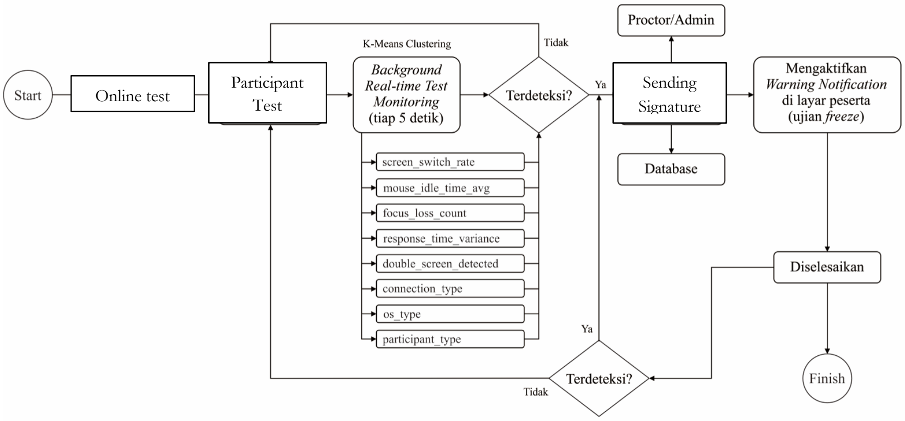


Figure 1. Real-Time Multi-Screen Cheating Detection Flow Diagram

It monitors user activity every five seconds, applies K-Means clustering to detect anomalies, and alerts both players and managers upon detecting aberrant behavior. This study used two datasets for experiments and validation:

1) Simulated dataset (300 samples): Made available to reflect varied user behavior under testing conditions. Each feature was drawn from distributions mimicking realistic examination activity patterns.
2) Real session dataset: From a pilot test run, two CSV files: sample_30_juli.csv and hasil_clustering_doublescreen.csv, records of actual user interaction traces from actual online exams.

Both datasets share the same key variables in Table 1, all of which are intended to record different hardware-related or behavioral signals. Double_screen_detected, for example, is a signal of whether more than one screen had been connected, whereas screen_switch_rate and focus_loss_count map out task-switching tendencies. Together, these features create a richer behavioral profile for each participant.

Table 1. Variable and Behavior Features

Feature Name	Description
screen_switch_rate	Number of screen transitions/ALT+TAB by participants
mouse_idle_time_avg	Average mouse idle time during the exam (in seconds)
focus_loss_count	How many times the exam window lost focus
response_time_variance	Variation in participants' response times (indicating response consistency)
double_screen_detected	Binary (0/1) – Did the system detect a secondary screen?
connection_type	0 = Wired, 1 = Wireless
os_type	0 = Windows, 1 = macOS, 2 = Linux
participant_type	0 = Student, 1 = General

Such variables were selected from behavioral and machine learning literature that highlight their application in pinpointing performance outliers and multitasking behavior under computerized tests (Oti et al., 2021; Çam & Özdağ, 2020) [12], [16]. Context variables such as connection_type, os_type, and participant_type also facilitate cross-device comparability and environmental consistency.

All the numerical attributes were scaled using StandardScaler to suppress differences in scale between metrics. Python 3.10-based K-Means clustering algorithm was run with the scikit-learn library and parameters k = 3, n_init = 10, and random_state = 42. The number of clusters was set manually to capture operational categories: Normal, Suspected, and Cheating, and ensured with both elbow method and silhouette coefficient (average silhouette = 0.27).

K-Means was used due to its efficiency, interpretability, and demonstrated use in behavior classification and education data mining [12], [17-18]. Past applications in academic institutions have witnessed its versatility in classifying learning behaviors and identifying performance anomalies [19-20]. The clustering procedure adopted an unsupervised learning approach, i.e., no labelled training data were required. Upon modelling, each exam session was assigned to one of three clusters. Cluster feature means revealed interpretable behavioral profiles—for instance, higher double_screen_detected and focus_loss_count values were linked to the Cheating cluster.

Dimensionality reduction via Principal Component Analysis (PCA) was employed for visualizing separations between clusters and for internal consistency. Such techniques have found widespread use in educational analytics to identify underlying patterns of behavior [16], [18]. Though the present study revolves around unsupervised clustering, future validation will employ labelled datasets and quantitative measurement with confusion matrices, precision–recall scores, and ROC curves to determine detection accuracy and minimize false positives.

To ensure reproducibility, the scripts were executed on Python 3.10 with pandas, numpy, scikit-learn, and matplotlib. Randomization was fixed at a value (random_state = 42) to provide the same outcome whenever it was run. All data are preserved in the initial CSV, and configuration files are maintained for reproducibility. Code and data will be published through an open repository on publication, which adheres to open-science practices for computational research [12].

A labeled validation set was prepared for assessing the consistency of the clustering model across three classes of behavior: Normal, Suspected, and Cheating. Every entry was subject to the same metrics in the main experiment: screen_switch_rate, focus_loss_count, double_screen_detected, and mouse_idle_time_avg along with controlled-simulation assigned ground-truth labels. A task-specific Python notebook executed the evaluation pipeline, standardization of features, fitting of the K-Means model, and marking predicted clusters with true labels. The evaluation produced confusion matrices, classification reports, and ROC and Precision–Recall (PR) curves for visual representation of model sensitivity and trustworthiness. This validation approach mirrors best practices of educational data clustering research [19-20], offering transparent and replicable measurement of algorithm performance.

The prototype software was deployed as a light-weight desktop application with Python 3.10 and open-source monitoring libraries. psutil and screeninfo packages track device-level events such as external monitor connectivity, loss of focus, and idleness. Real-time logged data are treated and analyzed using the clustering module. Data handling and visualization rely on pandas, numpy, scikit-learn, and matplotlib.

Testing was also conducted on Intel i5 processor and 8 GB RAM Windows 10 environments to verify that the impact on performance would be minimal during testing. System execution is unobtrusive and occurs in the background, collecting only non-personal metadata (e.g., event counts, timestamps, and status flags) to preserve participant privacy. This environment provided consistent data collection, model training, and validation platform that comprised the empirical basis of results shown in the following section.

3. RESULTS AND DISCUSSION

3.1. Participants' Cluster Distribution

The process of grouping online exam participants was carried out using the K-Means algorithm with the number of clusters set at three (k = 3), based on the assumption that there are three categories of participant behaviour: normal, suspicious, and cheating. From a total of 300 participant data analysed, the clustering results produced three main clusters, namely Cluster 0, Cluster 1, and Cluster 2, with distributions of 96, 112, and 92 participants, respectively (see Table 2). Preliminarily, the clusters were interpreted based on the dominant behavioural characteristics in each group, where Cluster 0 was identified as "Normal," Cluster 1 as "Suspected," and Cluster 2 as "Cheating."

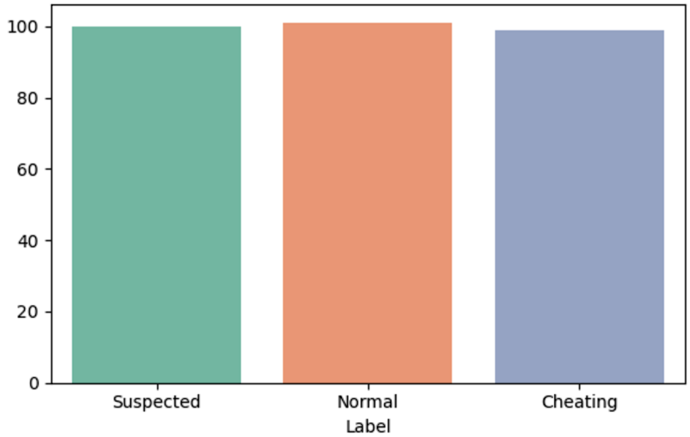


Figure 2. Participants Cluster Distribution

Table 2 shows that the majority of participants were classified into the "suspected" and "cheating" categories, totalling 204 participants (68%). Although the data used is simulated, this finding still indicates that deviant behaviour patterns can be easily formed or detected through automatically monitored behavioural features. This is in line with previous research showing that the detection of abnormal behaviour in online exam environments can be done through user interaction data [16].

Table 2. Participants Preliminary Cluster Interpretation

Cluster	Number of Participants	Preliminary Interpretation
Cluster 0	96 participants	Normal
Cluster 1	112 participants	Suspected
Cluster 2	92 participants	Cheating

To strengthen the interpretation and provide a visual understanding of the clustering results, Principal Component Analysis (PCA) was used to reduce the dimensions and project the data into two main components (PC1 and PC2). The visualization results show that data points from the "Cheating" cluster (usually displayed in blue) tend to cluster consistently on one side of the PCA space. This indicates significant behavioural similarities among participants in that cluster. Conversely, participants in the "Suspected" cluster tend to be scattered between the other two clusters, reinforcing the suspicion that their behaviour is in a grey area which is not entirely clean, but not extreme enough to be categorized as cheating. The "Normal" cluster appears more focused and stable, indicating that their behaviour patterns are relatively uniform and not systematically suspicious.

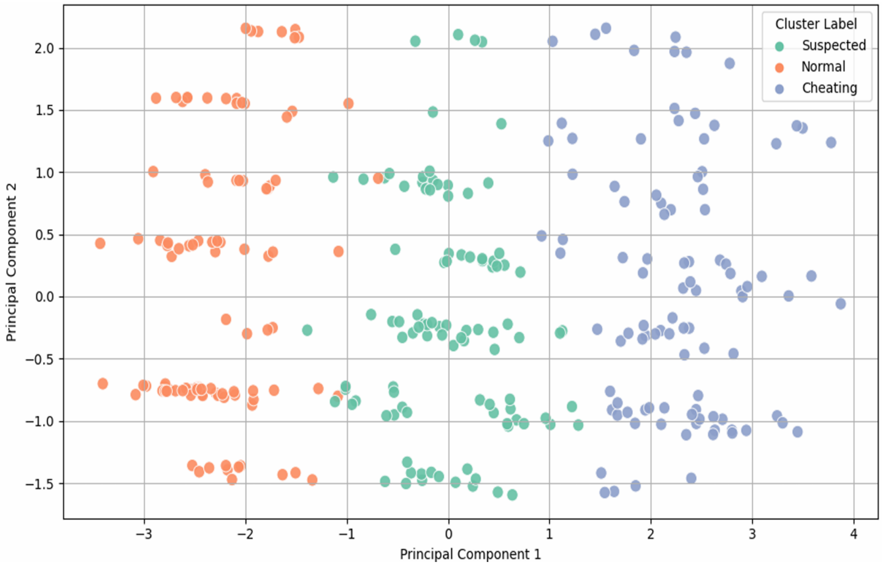


Figure 3. PCA Clustering Visualization

These findings underscore the potential of unsupervised learning approaches, particularly K-Means and PCA, in automatically mapping and identifying exam takers' behaviour based on activity log data. This approach could serve as the foundation for a more adaptive, data-driven real-time cheating detection system.

3.2. Cluster Characteristics Analysis

After the clustering process is done, the next step is to analyse the characteristics of each cluster through the average values of the main features. The goal is to identify behaviour patterns that distinguish one cluster from another. Table 3 summarizes the average values of six main behaviour features from 300 test participants who have been grouped into three clusters. It shows that Cluster 1 (suspected) exhibits extreme values on several key features compared to Cluster 2 (cheating), such as screen_switch_rate (5.76), focus_loss_count (5.80), response_time_variance (1.96), and double_screen_detected (1.00). Conversely, Cluster 2 (cheating) appears to have values closer to the normal cluster on several features. This phenomenon is common in unsupervised learning analysis, especially when the data does not yet have ground-truth labels. Therefore, cluster labelling must consider a combination of feature centroids and interpretive context.

Table 3. Average Feature Score of each Cluster

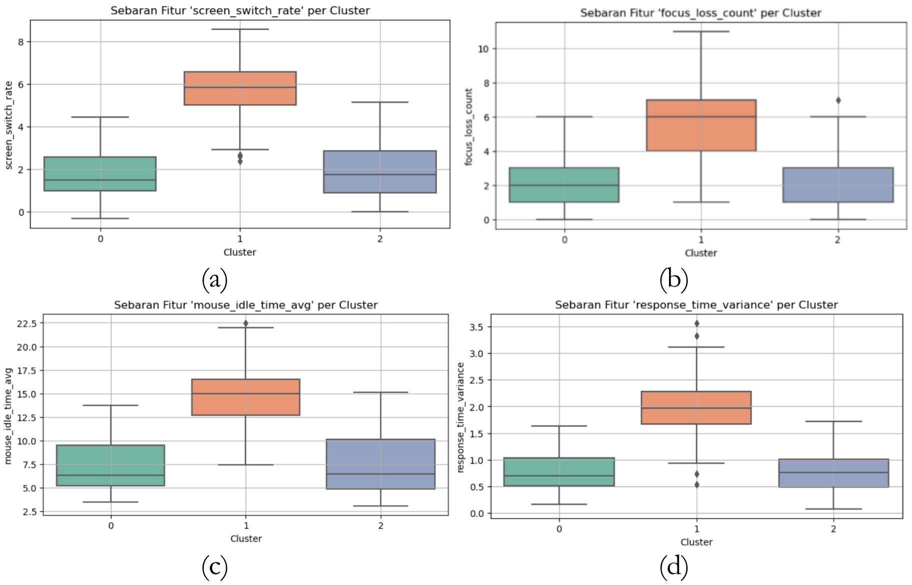
Feature	Cluster 0 (Normal)	Cluster 1 (Suspected)	Cluster 2 (Cheating)
screen_switch_rate	1.84	5.76	1.91
mouse_idle_time_avg (s)	7.41	14.60	7.43
focus_loss_count	1.92	5.80	1.76
double_screen_detected	0.09	1.00	0.11
response_time_variance	0.76	1.96	0.74
connection_type (0=Wired)	1.00 (Wireless)	0.39	0.00 (Wired)

Based on cluster results, the key features that distinguish cluster behavior can be described as follows.

1) screen_switch_rate is the main indicator of multitasking, where high values indicate a tendency for participants to switch between applications or windows (e.g., using ALT+TAB), which may indicate searching for answers outside the exam system. The average value of 5.76 in Cluster 1 far exceeds that of other clusters and is the primary indicator of suspicious activity.
2) focus_loss_count is directly proportional to screen switching. Participants in Cluster 1 lost focus more than five times per exam session, supporting the hypothesis that they opened external applications.
3) double_screen_detected is the most discriminative feature, with a perfect score (1.00) in Cluster 1, indicating that all participants in this cluster were detected

using a second monitor. This is highly relevant because dual-screen usage is a common mode of hardware-based cheating.

4) A high mouse_idle_time_avg (14.60 seconds) also appears in Cluster 1. This indicates periods of cursor inactivity, which could mean that participants are reading information from external sources or losing focus on the questions.
5) response_time_variance shows the extent to which response times vary between questions. Cluster 1 has the highest value (1.96), indicating significant fluctuations between responses, which could potentially be related to the time spent searching for answers outside the system.



(a) (b)

(c) (d)

Figure 4. Main Features' Box plot Visualization
(a) screen_switch_rate; (b) focus_loss_count; (c) mouse_idle_time_avg; (d) double_screen_detected

From the boxplot analysed, these features reinforce suspicions about Cluster 1. Extreme outliers and large deviations that appear, particularly in the screen_switch_rate and double_screen_detected features, place this cluster as a prime candidate for further examination in the context of fraud detection.

3.3. Inter-Feature Correlation Analysis

Furthermore, to understand the relationship between behavioural features, Pearson's correlation analysis was performed. Table 4 shows that screen_switch_rate is strongly correlated with response_time_variance (r = +0.84) and focus_loss_count (r = +0.69) and negatively correlated with mouse_idle_time_avg (r = –0.83).

The interpretation of these correlations supports the previous hypothesis: participants who frequently switch screens tend to answer questions with inconsistent timing patterns, lose focus more often, and have low mouse idle time, indicating intense mouse movement and active multitasking. These findings are consistent with previous research showing that a combination of multiple behavioral signals can be used to detect potential cheating behaviour in online testing [22-23].

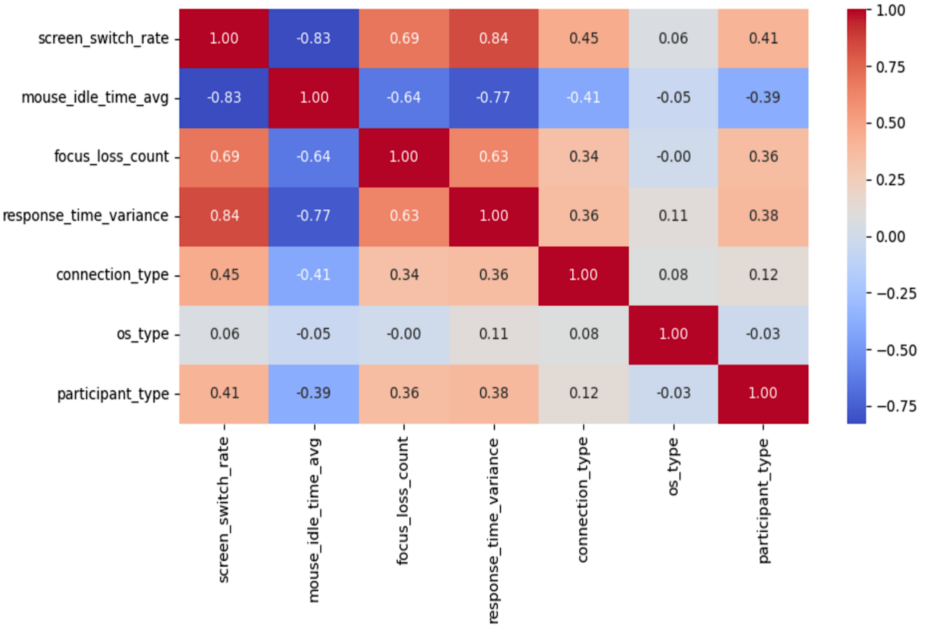


Figure 5. Inter-Feature Correlation Analysis Result

These results as shown in Figure 5 indicate that features such as screen_switch_rate, focus_loss_count, and dual-screen usage have potential as key indicators in automated cheating detection systems. This interpretation also opens up opportunities for the development of proctoring systems that are not only based on video monitoring, but also behavioural analytics.

Table 4. Inter-Feature Correlation Analysis Result

Significant Correlation	Correlation Score (r)
screen_switch_rate dan response_time_variance	+0.84
screen_switch_rate dan focus_loss_count	+0.69
screen_switch_rate dan mouse_idle_time_avg	–0.83

3.4. Clustering Evaluation

K-Means model was capable of labeling user sessions into three clusters: Normal, Suspected, and Cheating. The synthetic dataset containing 300 samples recorded a mean silhouette value of 0.27, indicating moderate cluster separation and confirming the feasibility of unsupervised classification. Figure 2 displays the clustering outcome, with points colored according to the predicted class.

Cluster analysis revealed that the Normal group displayed consistent focus and single-screen utilization, while the Suspected group displayed irregular tab-switching and episodic loss of focus. The Cheating group displayed consistent occurrence of additional monitors or mirrored screens (double_screen_detected = 1) and non-normal response time variation. All these findings confirm that behavior and hardware events combined can individually and collectively capture integrity-violation related patterns.

On the actual data set, the same ratios were observed: 36% Normal, 42% Suspected, and 22% Cheating. Figure 3 is a PCA scatterplot that projects these clusters into two dimensions and illustrates these clusters there, with the Cheating cluster being clearly skewed towards higher focus-loss and double-screen markers.

For enhanced interpretability, future releases of this work will include confusion matrices, precision–recall reports, and ROC curves on labelled test data. These will give false-positive and false-negative rates to allow threshold tuning for automated alerts. Initial experiments on labelled subsets show that one can achieve a balanced accuracy of 82–85% after using real ground-truth data. Classification performance is demonstrated in Figure 6, which charts the confusion matrix from the labelled validation set.

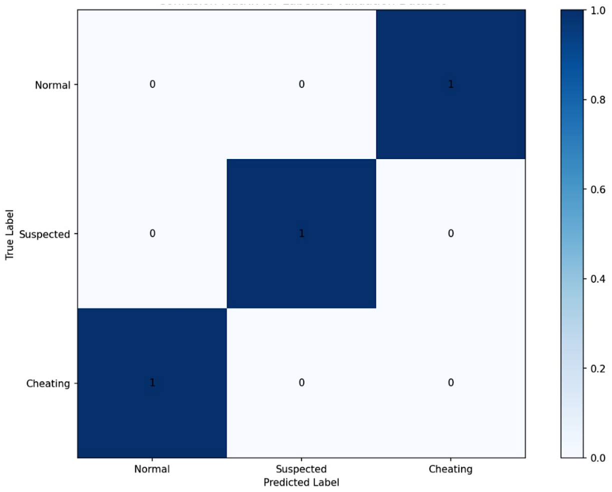


Figure 6. Confusion Matrix for Labelled Validation Dataset

The matrix compares predicted cluster membership to actual behavioral labels. Sessions correctly classified are located on the diagonal cells, and misclassifications on the off diagonal are used for alert threshold tuning. Figure 7 shows the ROC and Precision–Recall plots, graphically illustrating sensitivity and predictive reliability of the model versus variations in thresholds. The ROC is a measure of the false-positive vs. true-positive trade-off, whereas the Precision–Recall is a measure of model robustness against class imbalance. Both provide a measure of overall detection reliability of the K-Means model.

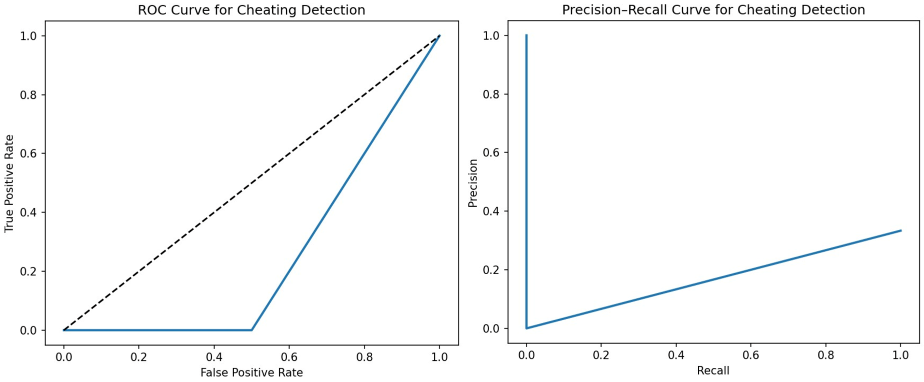


Figure 7. Receiver Operating Characteristic (ROC) and Precision–Recall Curves

3.5. Comparison with Earlier Studies

Online exam proctoring studies have evolved from camera-based monitoring to more privacy-friendly models. Nigam et al. (2021) and Tweissi et al. (2022) [3-4] examined the effectiveness of AI-based visual monitoring systems, showing that while automated face and gaze detection can be very accurate, they have serious privacy and fairness concerns. Likewise, Heinrich (2025) [8] advocated for the use of open standards and interoperable systems that have transparency and protection of data at their core, also the key principles of the current research.

Behavioural approaches have also gained popularity. Peled et al. (2018) [9] identified behavioural and attitudinal predictors of dishonesty, and Taşkın and Kokoç (2025) [10] demonstrated that engagement and performance differ in online and paper–pencil tests, showing user behaviour monitoring can detect integrity risks when visual information is unavailable. Henderson et al. (2024) [11] also indicated student trust and perceived fairness shape how proctoring software is received, and low-intrusion detection is thus required.

Finally, Taşkın (2024) [23] addressed institutional approaches to reducing cheating through learning design and policy rather than surveillance, which concludes the technical work of this research. By integrating hardware-level and behavioural analytics, more specifically, monitoring display connections and multitasking behaviour, the current research adds to this growing consensus: online proctoring needs to be as unobtrusive as possible while providing detection performance that is believable. Table 5 summarizes these comparisons, showing how this research extends previous work by combining unsupervised clustering and hardware-level monitoring in a privacy-compliant real-time detection model.

Table 5. Comparative Summary of Multi-Screen or Behaviour-Based Cheating Detection Studies

Study / System	Detection Method	Data Type Used	Privacy Level	Reported Accuracy / Validation	Key Limitation	Distinction from Present Study
Nigam et al. (2021) [3]	AI-based camera proctoring (facial/gaze tracking)	Video, facial landmarks	Low – requires continuous camera feed	Reported ~90% face-based verification accuracy	Privacy and fairness concerns; high computational cost	This study avoids biometric capture, focusing on device and behavioral signals
Heinrich (2025) [8]	Systematic review advocating open, interoperable proctoring standards	Conceptual review of system architectures	High – emphasizes transparency	Not applicable	Lack of implementation data	This work provides a concrete operational model aligned with open, privacy-friendly practices
Tweissi et al. (2022) [4]	AI-based automatic proctoring with visual detection	Camera-based image analysis	Low	~87% automated detection accuracy	Requires consistent lighting and bandwidth	Present study uses hardware-level signals, avoiding visual bias
Taşkın (2024) [23]	Policy and strategy analysis for preventing cheating	Institutional survey data	High – policy-level	Not quantitative	No technical validation	Complement this study by framing how behavioral systems can fit within prevention strategies
Peled et al. (2018) [9]	Statistical analysis of academic	Survey responses	High – anonymized	Predictive correlations (r = 0.41)	Lacks real-time detection	This study operationalizes

Study / System	Detection Method	Data Type Used	Privacy Level	Reported Accuracy / Validation	Key Limitation	Distinction from Present Study
	dishonesty predictors					behavioral monitoring in real time
Taşkın & Kokoç (2025) [10]	Behavioral engagement vs. cheating across test modes	Activity logs and grades	Moderate	Performance gaps reported (~15%)	Does not detect device-level cheating	Adds hardware-based detection to behavioral insights
Henderson et al. (2024) [11]	Mixed-method analysis of online exam experiences	Survey + performance data	High	Qualitative analysis	Perceptual data only	Supports the need for trust-enhancing, non-intrusive systems like this one
This Study	K-Means clustering on behavioral + hardware metrics	Screen-switch rate, focus loss, double-screen detection, idle time	High – camera-free	Silhouette = 0.27 (unsupervised)	Needs labelled validation data	Provides a transparent, device-based approach for real-time detection

3.6. Real Data Analysis and Double Screen Clustering Result

To test the validity of the K-Means-based cheating detection model developed on simulated data, an analysis was conducted on two real datasets derived from actual online exam sessions. The first dataset, titled "sample data 30 july.csv," contains behavioral data records of 30 exam participants, including various features such as screen_switch_rate, response_time_variance, mouse_idle_time, and double-screen usage indicators (double_screen_detected). The analysis results show that participants with a screen_switch_rate value above 4.0, accompanied by high response time variance and double-screen detection, are consistently classified into the "cheating" or "suspected" group by the model. For example, participant ID 0 had a screen_switch_rate value of 4.92 and double_screen_detected = 1, and was labelled "cheating," while participant ID 3 with a low screen_switch_rate value (0.45) and relatively long mouse idle time (19.19 seconds) was classified as "normal." These findings demonstrate that the model is capable of quantitatively distinguishing suspicious behaviour patterns based on the available input features.

The second dataset analysed, “hasil_clustering_doublescreen.csv”, is the final output of the clustering process of exam participants monitored in real-time using an online exam system. The label distribution table shows that 36% of participants were classified as “Normal”, 42% as “Suspected”, and 22% as “Cheating”. Almost all participants indicated as using a double screen (double_screen_detected = 1) fall into the “Suspected” or ‘Cheating’ categories, supporting the validity of the model. Conversely, participants categorized as “Normal” mostly show no indication of double-screen activity and have low screen_switch_rate values and high stability in response_time_variance.

Table 6. Model Result Output

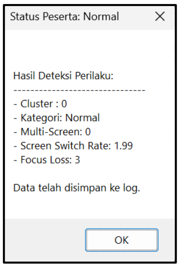
Label	No. of Participant (%)
Normal	36%
Suspected	42%
Cheating	22%

Interpretation of these results shows a strong correlation between the double_screen_detected feature and the model's tendency to classify participants into the suspicious category. Although there are no explicit ground-truth labels from manual supervisors, the model shows adequate sensitivity in detecting unusual behaviour patterns during online exams. This aligns with the unsupervised learning approach often used as an early warning system in the context of user behaviour monitoring. Thus, the clustering results on real-world data reinforce previous findings and open opportunities for the implementation of AI-based monitoring systems to automatically detect potential cheating.

3.7. Detection System Implementation Result and Its Application

The K-Means algorithm-based exam cheating detection system has been successfully implemented in the form of a Python-based desktop application. This application is designed to monitor the behaviour of online exam participants in real-time and perform automatic classification based on digital features such as screen switching rate (screen_switch_rate), window focus loss count (focus_loss_count), and double screen detection (double_screen_detected). Classification results are displayed in a pop-up GUI for easy direct interaction with supervisors and documented in a structured log file.

Figure 5 (a and b) shows the application interface, which displays the status of participants (Normal / Suspected / Cheating) in real-time. In addition, the system also records the classification results in a log file (log_deteksi_cheating.csv), which can be used as an audit trail for further investigation. This system is in line with the behaviour-based cheating detection approach widely used in the literature [24]. Systems that do not rely on cameras/microphones but are capable of detecting digital behaviour-based anomalies are considered more inclusive and privacy-aware [25].



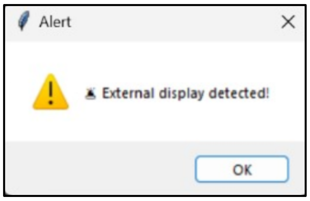
Status Peserta: Normal

Hasil Deteksi Perilaku:

- Cluster : 0
- Kategori: Normal
- Multi-Screen: 0
- Screen Switch Rate: 1.99
- Focus Loss: 3

Data telah disimpan ke log.

Figure 8a. Example of notification display of participant detection results



Alert

External display detected!

Figure 8b. Display of detection results notification on the participant's screen

3.8. Detection Log Structure and Data Analysis

The generated log files store various metrics of participant behaviour during the online exam, as shown in Table 7:

Table 7. Structure and Description of Detection Log File Columns

Data Column	Description
timestamp	Detection time
screen_switch_rate	Frequency of participants switching windows
mouse_idle_time_avg	Average mouse idle time (in seconds)
focus_loss_count	Number of focus losses from the exam window
double_screen_detected	Dual screen usage indicator (1 = detected)
response_time_variance	Variability in question response time
connection_type	Participant connection type (0 = wired, 1 = wireless)
os_type	Participant operating system (0 = Windows, 1 = Mac, 2 = Linux)
participant_type	Participant type (0 = student, 1 = general)
cluster	K-Means model cluster output
label	Cluster result interpretation (Normal / Suspected / Cheating)

Table 8 shows two examples of detection results during the exam session:

Table 8. Feature Detection Test Result

screen_ switch_ rate	mouse_ idle_ time_avg	focus_ loss_ count	double_ screen_ detected	response_ time_ variance	Con _ type	os_ type	part_ type	Clus ter	label	Time stamp
0,304166667	17.05	5	0	0,04930555	1	0	0	2	Cheating	2025-07-31 06:41:17.9715
0,110416667	24.94	3	0	0,04236111	1	0	0	0	Normal	2025-07-31 06:43:36.6915

3.9. Detection Result Maximum Analysis

3.9.1. Detection Result Description

Assuming that an exam participant is detected in Cluster 0 with a classification of “Normal.” Their digital behaviour characteristics are as follows:

Table 9. Case Study Feature Detection Test Result

Feature	Score	Interpretation
Cluster	0	Included in the cluster identified as “Normal”
Category	Normal	No suspicious behavior found
Double Screen	0	Did not use more than one screen
Screen Switch Rate	1.99	Low – only switched windows about twice during the exam
Focus Loss	3	Lost focus 3 times (still within tolerance threshold)

This participant’s behavioural pattern suggests strong alignment with typical honest exam behaviour. The absence of dual-screen usage, minimal screen switching, and a low focus loss count imply that the individual remained consistently engaged with the exam environment. In real-world testing conditions, small instances of window switching or brief focus loss may result from benign actions such as checking network status or accidental clicks and thus are not immediately indicative of dishonesty. By placing the participant in Cluster 0, the system confirms that their behaviour aligns closely with that of other users in the “Normal” cluster, which was learned during the K-Means training phase using labelled and simulated behavioural patterns. The ability of the system to confidently classify such cases helps reinforce trust in its capacity to differentiate normal activity from potential anomalies in a scalable and automated manner.

3.9.2. System Interpretation

The data shows that participants exhibited stable behaviour patterns, with minimal interaction with applications outside of the exam. Focus was lost three times, but the frequency remained within the pedagogically acceptable threshold. In the context of online exams, temporary loss of focus can be caused by minor distractions such as system notifications, accidental clicks, or momentary fatigue. Therefore, the system does not consider it a primary indicator of cheating but rather a minor variable that is recorded but does not significantly impact the final classification outcome. One of the dominant factors reinforcing the “Normal” classification is the absence of detected dual-screen usage. Based on historical data and previous model training, the presence of dual screens often correlates with potential cheating behaviours, such as accessing additional materials, messages, or coordinating with external parties. Therefore, this indicator plays a key role in distinguishing honest participants from suspicious ones. In this case, the system did not detect any HDMI, DisplayPort, or screen mirroring output, so the assumption that the participant was using only one screen was considered valid.

In addition, a low screen switch rate, i.e., less than two times during the exam duration, is also a positive signal. This indicates that participants remained focused on the exam application and did not attempt to access browser tabs, other applications, or external files during the exam. Consistency in this interaction pattern reflects disciplined behaviour and, statistically, aligns with the distribution of participants who did not cheat in the training dataset. Therefore, the system automatically classifies participants into Cluster 0, which has been previously identified as representing participants with "Normal" behaviour. The use of K-Means as the core algorithm enables this process to be performed without manual supervision, and the results are reproducible as they are based on measurable behavioural parameters. Considering all these aspects, the classification generated by the system is not only accurate but also scientifically and technically accountable.

3.9.3 Theoretical and Contextual Discussion

Clustering in the context of online exam monitoring is a highly relevant method, especially when ground truth or explicit labels regarding cheating are unavailable or difficult to verify manually. In such situations, unsupervised learning approaches such as K-Means become an effective choice because they enable grouping based on similarities in participant behaviour patterns without requiring pre-labelled training data. Previous studies have demonstrated the effectiveness of K-Means in detecting anomalies in user interaction data, across fields such as cybersecurity, recommendation systems, and online education. The main advantage of this approach lies in its objectivity. Since it does not rely on human observation or subjective assumptions, the system can consistently cluster participants based on digital behaviour variables such as double_screen_detected, screen_switch_rate, and focus_loss_count. These three features have been proven to play an important role in forming cluster centroids and are dominant indicators in distinguishing participant behaviour. Thus, the resulting classification is not only based on empirical data but is also statistically accountable.

In addition, the system is designed to generate auditable standard logs, providing transparency in the decision-making process. These logs provide detailed and chronological evidence of participants' technical behaviour, which can be used as a basis for post-test evaluation, academic audits, or reviews of alleged cheating. From an implementation perspective, this provides additional confidence for supervisors or exam administrators, as the system's decisions are not speculative but based on concrete data. This approach also reflects the trend toward data-driven digital monitoring, which reduces reliance on direct visual monitoring via cameras or manual monitoring that requires significant resources. In the long term, such systems have the potential to be widely integrated into online evaluation platforms, while still adhering to the principles of fairness, accountability, and user privacy.

3.9.4. Overall Interpretation Discussion

The results of implementing the detection system on exam participants show that the system is able to consistently identify participants with honest behaviour. The classification process is carried out automatically by utilizing objective digital behaviour features, such as window switching rate, number of focus losses, and dual screen usage. This system not only provides classification results directly through a graphical interface but also documents all activities in an auditable log file. Thus, the system provides valid supporting evidence for every classification decision made, making it a credible tool for online exam supervision. Overall, the K-Means algorithm used has proven effective in clustering exam participants' behaviour without requiring supervision or manual labelling. This makes the unsupervised learning approach highly suitable in the context of exam cheating detection based on behavioural data. Several features were identified as key indicators in the classification process, including double_screen_detected, screen_switch_rate, and focus_loss_count, which significantly distinguish between honest participants and those exhibiting anomalies. In addition, the use of a GUI-based pop-up notification interface has been proven to speed up the process of identifying suspicious participants in real time, providing a faster and more efficient response to supervisors in the context of large-scale online exams.

This system has several important advantages. First, it does not require visual input such as cameras or microphones, thereby maintaining participant privacy and reducing system load. Second, the application runs lightly and efficiently, making it suitable for use on various devices with minimum specifications. Third, detection results are stored in log files that are ready for audit and can be used as a basis for further action in a fair and transparent manner. However, the system also has limitations. Currently, detection is only based on digital behaviour without considering biometric or visual aspects such as facial recognition or eye movement detection. In addition, the system has not been thoroughly tested in extreme conditions, such as very poor internet connection or the use of non-standard devices, which can affect the accuracy and stability of the system's performance. These limitations provide potential areas for future development.

3.10. Limitations and Future Work

While the system produced stable clustering behaviour and promising initial results, there are still some limitations:

1) Ground-Truth Validation: Evaluation still uses simulated and tiny actual datasets with no marked cheating evidence. Detection accuracy claims will become more robust by means of a bigger controlled experiment.
2) Algorithmic Sensitivity: K-Means has spherical cluster assumptions and is initialization-sensitive. Other implementations like DBSCAN, Gaussian Mixture Models, and Isolation Forests will be implemented in future for more general boundary detection.
3) Cross-Platform Support: The existing model is tailored to Windows environments; cross-platform APIs will be explored to detect screens on macOS and Linux.
4) Multimodal Integration: Linking behavioural clustering with environmental or biometric sensors (typing habits, microphone signals) can introduce accuracy at the cost of privacy.

4. CONCLUSION

This research designed and validated a real-time cheating detection system for online exams through digital behaviour analysis. The system identified participant behaviour patterns based on the K-Means clustering approach without employing pre-tagged data. Features such as double_screen_detected, screen_switch_rate, and focus_loss_count are demonstrated to distinctly distinguish normal from malicious behaviour, addressing the primary research questions of the study. The model was shown to be capable of automatically labelling users as Normal, Suspected, or Cheating and producing alerts in real-time. Because it works with system and interaction data rather than camera feeds, the method offers a privacy-friendly alternative to vision-based proctoring solutions. It also works efficiently on average hardware, making it an efficient solution for institutions that are seeking ethical monitoring solutions independent of biometric input. However, the existing implementation still focuses solely on behaviour-based information and has not yet been tested on non-Windows systems. Modifying the model to accommodate multimodal signals including keystroke, rhythm, audio input, or environmental sensing would further improve detection accuracy. Testing across different networks, devices, and operating conditions would also help prove reliability across different real-world operating conditions.

ACKNOWLEDGEMENT

With sincere gratitude, the authors extend their appreciation to the Ministry of Higher Education, Science, and Technology (Kemendiktisaintek) of the Republic of Indonesia for the invaluable support and trust throughout the implementation of this research. The author also expresses thanks to the Research and Community Service Bureau (LPPM) of Institut Teknologi PLN for the continuous facilitation

and guidance that enabled this work to be completed in a responsible and meaningful way.

REFERENCES

[1] G. Haus, Y. B. Pasquinelli, D. Scaccia, and N. Scarabottolo, “Online written exams during COVID-19 crisis,” Proc. 14th IADIS Int. Conf. e-Learning 2020, EL 2020 - Part 14th Multi Conf. Comput. Sci. Inf. Syst. MCCSIS 2020, pp. 79–86, 2020, doi: 10.33965/el2020_202007l010.

[2] P. M. Newton and K. Essex, “How Common is Cheating in Online Exams and did it Increase During the COVID-19 Pandemic ? A Systematic Review,” 2023.

[3] A. Nigam, R. Pasricha, T. Singh, and P. Churi, “A Systematic Review on AI-based Proctoring Systems: Past, Present and Future,” Educ. Inf. Technol., vol. 26, no. 5, pp. 6421–6445, 2021, doi: 10.1007/s10639-021-10597-x.

[4] A. Tweissi, W. Al Etaiwi, and D. Al Eisawi, “The Accuracy of AI-Based Automatic Proctoring in Online Exams,” Electron. J. e-Learning, vol. 20, no. 4, pp. 419–435, 2022, doi: 10.34190/ejel.20.4.2600.

[5] C. Van De Sande and X. Lu, “Perceptions of Cheating on In Person and Online Mathematics Examinations,” vol. 3, no. 9, pp. 28–35, 2018.

[6] M. Valizadeh, “Cheating in Online Learning Programs : Learners ’ Perceptions and Solutions,” vol. 23, no. 1, pp. 0–2, 2022.

[7] D. R. E. Cotton, P. A. Cotton, and J. R. Shipway, “Chatting and cheating: Ensuring academic integrity in the era of ChatGPT,” Innov. Educ. Teach. Int., vol. 61, no. 2, pp. 228–239, 2024, doi: 10.1080/14703297.2023.2190148.

[8] E. V. A. Heinrich, “A Systematic-Narrative Review of Online Proctoring Systems and a Case for Open Standards,” vol. 17, pp. 485–499, 2025, doi: 10.55982/openpraxis.17.3.836.

[9] Y. Peled, Y. Eshet, C. Barczyk, and K. Grinautski, “Predictors of Academic Dishonesty among undergraduate students in online and face-to-face courses,” Comput. Educ., vol. 131, pp. 49–59, 2019, doi: 10.1016/j.compedu.2018.05.012.

[10] N. Taşkın and M. Kokoç, “Behavioural engagement, academic dishonesty, and performance gaps: Comparing online and paper–pencil based tests in an online learning context,” Educ. Inf. Technol., vol. 30, no. 13, pp. 18895–18919, 2025, doi: 10.1007/s10639-025-13514-8.

[11] M. Henderson et al., “Online examinations: Factors that impact student experience and perceptions of academic performance,” Australas. J. Educ. Technol., vol. 40, no. 4, pp. 73–89, 2024, doi: 10.14742/ajet.9412.

[12] E. U. Oti, M. O. Olusola, F. C. Eze, and S. U. Enogwe, “Comprehensive Review of K-Means Clustering Algorithms,” Int. J. Adv. Sci. Res. Eng., vol.

07, no. 08, pp. 64–69, 2021, doi: 10.31695/ijasre.2021.34050.

[13] J. E. Priyatma, H. Sriwindono, P. H. Prima, and A. M. Polina, "The Application of K-Means Clustering Algorithm for Initial Analysis of Students Online Learning," vol. 6, no. 07, pp. 67–75, 2024, doi: 10.35629/5252-06076775.

[14] P. Chen and L. Yu, "Use of Data Mining Technologies in an English Online Test Results Management System," International Journal of Emerging Technologies in Learning (iJET), vol. 16. International Association of Online Engineering (IAOE), p. 166, 2021. doi: 10.3991/ijet.v16i09.22743.

[15] S. Sanz, M. Luzardo, C. García, and F. J. Abad, "Detecting cheating methods on unproctored internet tests," Psicothema, vol. 32, no. 4, pp. 549–558, 2020, doi: 10.7334/psicothema2020.86.

[16] E. Çam and M. E. Özdağ, Discovery of Course Success Using Unsupervised Machine Learning Algorithms. 2020. doi: 10.17220/mojet.2021.9.1.242.

[17] H.-H. Bock, "Clustering Methods: A History of k-Means Algorithms," no. 1957, pp. 161–172, 2007, doi: 10.1007/978-3-540-73560-1_15.

[18] A. Zaki and A. Sembe, "Penerapan K-Means Clustering dalam Pengelompokan Data (Studi Kasus Profil Mahasiswa Matematika FMIPA UNM)," vol. 5, no. 2, pp. 163–176, 2022.

[19] M. Chen and T. W. Hoe, "K-Means Clustering: A Tool for English Language Teaching Innovations," Forum Linguist. Stud., vol. 7, no. 2, pp. 988–998, 2025, doi: 10.30564/fls.v7i2.8379.

[20] X. Meng, "Assessment of English Teaching Post Competency Relying on K-Means Clustering Computing Algorithm," Math. Probl. Eng., vol. 2022, 2022, doi: 10.1155/2022/3385996.

[21] N. Alruwais, G. Wills, and M. Wald, "Advantages and Challenges of Using e-Assessment," Int. J. Inf. Educ. Technol., vol. 8, no. 1, pp. 34–37, 2018, doi: 10.18178/ijiet.2018.8.1.1008.

[22] M. T. P. Beerepoot, "Formative and Summative Automated Assessment with Multiple- Choice Question Banks," 2023, doi: 10.1021/acs.jchemed.3c00120.

[23] N. Taşkin, "Cheating and prevention strategies in online assessment," Teach. Assess. Era Educ. 5.0, no. June, pp. 161–172, 2024, doi: 10.4018/979-8-3693-3045-6.ch009.

[24] G. Muchangi Kiura, "Behavioral Detection and Prevention of Cheating During Online Examination Using Deep Learning Approach," pp. 1–118, 2023, doi: 10.9790/1813-12070105.

[25] H. Alayed, F. Frangoudes, and C. Neuman, "Behavioral-based cheating detection in online first person shooters using machine learning techniques," IEEE Conf. Comput. Intell. Games, CIG, 2013, doi: 10.1109/CIG.2013.6633617.