Journal of Computer Science

Original Research Paper

An Open-Source Online Examination System to Meet the Integrity Demands of E-Learning

Abdul Wahab Muzaffar

College of Computing and Informatics, Saudi Electronics University, Riyadh, 11673, Saudi Arabia

Article history
Received: 12-12-2023
Revised: 04-02-2023
Accepted: 08-03-2024

Email: a.muzaffar@seu.edu.sa

Abstract: The rise of online learning platforms and the growing demand for remote education emphasize the importance of online exam-proctoring tools. Online proctoring tools presented in the literature require high internet speed and specialized hardware support, posing accessibility challenges for individuals in developing countries. This study aims to develop a solution that relies on something other than high internet speed and high-end hardware components. The proposed solution extracts data generated from keystroke logs, browser history, and applications opened during the assessment to predict online exam cheating. This data is compared to the words in the test using Term Frequency (TF) and Inverse Document Frequency (IDF) to predict cheating. To evaluate the effectiveness of the proposed solution, an experiment was conducted with sixteen undergraduate Software Engineering students divided into two groups of eight students. The groups were given 20-minute-long software engineering and database exams, each comprising 30 MCQS. These exams were conducted with the proposed proctoring tool and only one group was allowed to cheat. Results indicated that the proposed tool effectively detects cheating during exams. This approach can mitigate the digital divide, particularly for individuals lacking high-speed internet access and costly hardware. Consequently, the study proposes an inclusive solution designed to cater to users from diverse demographic backgrounds.

Keywords: Proctoring, Natural Language Processing, E-Learning, Online Exam, Online Exam Cheating

Introduction

Education has witnessed a transformative evolution driven by technology in the digital age. The emergence of online learning platforms and the increasing demand for remote education opportunities have reshaped the landscape of modern education (Palvia *et al*., 2018). The increased adoption of online education during crises like the war and pandemics highlights the significance of online exam-proctoring tools. In light of the challenges faced by educational institutions in maintaining conventional, face-to-face instructional methods, the use of remote learning has become a crucial measure to guarantee the uninterrupted provision of education (Gudiño Paredes *et al*., 2021). During the COVID-19 epidemic, UNESCO reported that school closures impacted more than 1.5 billion students in about 190 countries, requiring a transition to remote and online instruction (Unesco, 2022). This shift has increased the demand for trustworthy and protected evaluation methods to ensure academic integrity in online learning environments. Online examination proctoring solutions have become essential in the current educational transition. These solutions utilize various technologies, including artificial intelligence and webcam monitoring, to enable secure and convenient remote examinations (Hussein *et al*., 2020).

While previous research studies have made significant advances in proposing innovative solutions for online exam proctoring tools, a critical oversight demands our attention. Many of these studies have predominantly concentrated on improving the effectiveness and security of these technologies while neglecting the worldwide digital gap that impedes their accessibility, particularly in socioeconomically deprived areas (Slusky *et al*., 2020). For instance, a significant percentage of the global population, comprising approximately 3 billion individuals from Africa, Asia, and oceanic countries, faces challenges, including poverty, meager income, and restricted availability of high-speed internet (UNDESA, 2022). These issues collectively limit individuals' capacity to interact with Information Communication Technologies (ICTs) effectively, the fundamental basis for

electronic learning and engagement in remote assessments. The dominant remote assessment proctoring solutions that use biometrics, visuals, and audio to detect exam fraud often require high internet speed to be effective in real time (Ngqondi *et al*., 2021). Besides, hardware that incorporates biometric assessment capabilities and visuals often pushes the price of the resulting ICT device beyond the reach of people based in economically underdeveloped countries. As noted, such populations can barely afford high-speed internet connectivity (UNDESA, 2022) While advancements in exam proctoring solutions presented in the literature are a positive development, they indirectly promote the digital divide between those who can and cannot afford such technologies. This gap highlights the need for a more inclusive approach to developing and implementing online exam proctoring systems, as it predominantly affects those in economically challenged areas.

To tackle these issues, it is necessary to reevaluate the assumptions underlying the deployment of proctoring solutions and commit to developing more accessible and cost-effective alternatives to bridge the digital divide. By doing so, we can ensure that the benefits of online education and remote assessment are accessible to all individuals, regardless of their location or socioeconomic status. This study suggests a remote exam proctoring solution based on natural language processing. Figure 1 gives an overview of the study. Students interact with the system through the exam user interface. First of all, the proposed approach collects data generated by the keystroke logs, open applications, and browser history during assessment and preprocesses it to make sure that the information used for analysis is accurate. The collected data is preprocessed or cleaned by removing irrelevant data, handling missing and duplicate information, and filtering out noise and outliers, ensuring that the data is free from any errors and inconsistencies. Once the data is cleaned properly, the feature extraction process extracts the data important for cheating prediction and matches it with exam questions using Term Frequency (TF) and Inverse Document Frequency (IDF). This data is used to calculate the average similarity per question to predict cheating. We used the normalized exponential function or softmax activation function to compute probabilities for each question. The average similarities and probability results are computed for the final cheating prediction. Based on the percentage of the data matched with the questions in the exam, the tool predicts results as fair or cheating. The instructor can view these results by interacting with the system through the Administration user interface and make further decisions to take action based on cheating prediction.

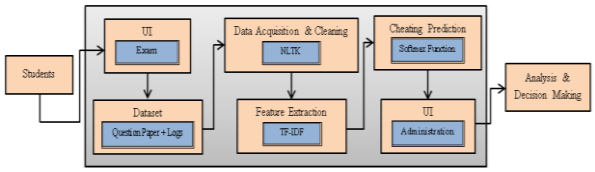

Fig. 1: Research overview

To evaluate the effectiveness of the proposed tool, sixteen students with software engineering degrees are randomly divided into two groups, each comprising eight students. The groups were given an online software engineering and databases exam, respectively. Each exam was 20 min long and had 30 MCQs. Group 1 students were instructed to act like exam takers. Group 2 students were allowed to cheat using the internet and other apps. The exams were conducted with the proposed proctoring tools and the results demonstrated that the tool can effectively predict cheating during exams. The proposed solution may significantly reduce reliance on high internet speed and other expensive hardware components to indicate real-time exam cheating.

Preliminaries

This section discusses the current state of the art and its limitations. Mainly, a comprehensive review of existing research, methodologies, and findings on online exam proctoring tools is presented. The key gaps and areas where further investigation is needed to address the challenges and limitations of current proctoring solutions are presented.

E-Learning

Education has experienced enormous transformations in the digital age, with e-learning at the forefront of this transition. E-learning has become a powerful and adaptable teaching method, utilizing digital technologies and the internet to deliver educational content and facilitate e-learning. It has caused a paradigm shift in acquiring knowledge and skills, providing accessibility, flexibility, and scalability that traditional learning methods frequently fail to provide. There has been an exponential rise in the use of e-learning-based education during the past decade, especially during war, pandemics, and natural calamities. e-learning has produced promising results during such critical times (Muzaffar *et al*., 2021).

Various e-learning-based platforms, including online learning management systems, are introduced and adopted worldwide. Students and educators can share their information and work together on these platforms, which are virtual libraries, classrooms, and work areas. They provide many tools, such as multimedia content, examinations, discussion boards, and means to get feedback. This creates an engaging learning atmosphere. E-learning platforms and management systems have

created enormous opportunities for distant lecture delivery, course and exam management, and student performance appraisal (McCoy *et al*., 2015).

Online Exam Cheating Solutions

Over time, E-learning-based platforms have been embedded with sophisticated and state-of-the-art image processing, machine learning, and other Artificial Intelligence (AI) based techniques so that universities and educational institutes adopt them reliably in a literal sense (Abisado *et al*., 2019). The foolproof online examinations are integral to E-learning for conducting fair and candid student appraisals (Andersen *et al*., 2020). Management of online examinations through such distant learning platforms is the most challenging task since the exams are conducted without the physical presence of students and proctors in the same venue. In contrast to the multifaceted enlightening opportunities that E-learning creates, examinations conducted in such settings are prone to cheating. The availability of enormous resources with online information may tempt students to cheat during online examinations. Without continuous monitoring, the integrity and reliability of the complete system can be questioned and, therefore, rejected by the didactic community (Muzaffar *et al*., 2021). Researchers have proposed various state-of-the-art approaches based on multiple technologies to ensure security, integrity, and reliability in online exams. The primary task of these techniques is to identify students' abnormal or suspicious behavior during the examination.

Various online proctoring solutions based on image processing, Natural Language Processing (NLP), and machine learning have been proposed by researchers. Fan *et al*. (2016) proposed an image-processing technique to analyze the cheating likelihood of examinees during online exams. Mainly, the authors utilized Kinect devices to capture the gestures of the examinee. Their algorithm starts with data preparation and then sampling three events from the gathered evidence. The sampled data is analyzed to establish the examinee's behavior. This leads to the prediction of exam cheating by the examinee during online exams. Although this proposed technique improved the prediction of exam cheating using multiple gestures, its applicability in a real online exam environment still needs to be investigated due to its limited scope. Similarly, Atoum *et al*. (2017) proposed an online exam proctoring system where online audio-visual streams of the examinee are analyzed to predict the cheating odds. To cover the broad physical area of the examinee, the data streams for processing are captured through two cameras and a microphone. The indicative features for cheating odds are extracted from audio-visual streams to achieve real-time proctoring. Subsequently, a Support Vector Machine (SVM) classifier is utilized to predict the cheating likelihood of the examinee. Experimental results showed high accuracy in predicting cheating. However, Atoum *et al*. (2017) solution requires higher-end hardware specifications and network resources to be effective.

Mohammed *et al*. (2023) proposed a novel e-exam cheating detection approach. This method uses IoT and Muse2 devices to detect the examinee's physiological condition and decide if it is “normal” or “abnormal” through EEG signals. Abnormal states can indicate cheating. CNN was used to determine the examinee's condition. The EEG signals of 15 volunteers aged 23-26 from the fourth stage of the computer engineering department/university of Mosul indicated a clear difference between “calm” or “normal” and “stress” or “abnormal” states. For many testing datasets, the system proved accurate. The dataset had two primary parts: 30 and 60 sec. The best accuracy was 97.37% for 30 and 97.14% for 60 sec. They found that the Muse2 device can reliably capture the EEG signal, which contains a lot of vital information that may be used to detect the examinee's physiological status. Kaddoura and Gumaei (2022) proposed a deep-learning approach to identify cheating in recorded video frames and speech. The developed method comprises three main parts: The speech-based detection module, the front camera-based cheating detection module, and the back camera-based cheating detection module. It uses deep Convolutional Neural Networks (CNNs) and the Gaussian-based Discrete Fourier Transform (DFT) statistical method to automatically extract usable visual pictures and voice characteristics to categorize and detect exam cheating. The proposed approach was evaluated by extensive experiments on a publicly available large-scale database, using various assessment metrics. Gopane and Kotecha (2022) proposed a method to validate and verify users continuously to maintain integrity. During the test, subtle micro-expression detection is performed, including laughter, eye gaze tracking to anticipate the applicant's gazing direction, blinking/close duration, and head activity/movement identification. Any act or moment of suspicion exhibited by the applicant is monitored and a penalty is imposed accordingly. The methodology employs artificial intelligence to categorize the applicant's behavior. The initial experimental findings exhibit the effectiveness of the suggested methodology.

Kasinathan *et al*. (2022) proposed prothorax, an automated online proctoring system that can track gaze, estimate head pose, detect faces, and monitor browser activity. Using smart facial detection algorithms, the suggested method immediately notifies an instructor of any suspicious movement or activity. Proctorex's primary benefit is its reliance on image processing technology, which allows instructors to view test candidates in real time. It is very easy for teachers and students to utilize and it doesn't require the involvement of a third party.

Instructors can use proctorex without going through a lengthy procedure and students can take an online exam without a drawn-out verification process. Sarmiento *et al*. (2023) examined how machine learning can detect academic dishonesty in online exams. User input device data was fed into logistic regression, Support Vector Machines (SVM), and random forest machine learning models. These components were then combined to build an ensemble model. The models were trained with two feature sets. The first set had four features; the second set had seven. The research found that models with more features performed better than those with four. The logistic regression model with standardization performed best, with 65% sensitivity and 87% specificity. (Garg and Goel, 2023) presented a machine learning-based model for detecting Internet cheaters by analyzing assessment log files. They modified an online quiz tool to collect tagged data. They extracted thirteen characteristics from the assessment log files' student and question features through feature engineering. They evaluated models created using ANOVA and Mutual Information feature selection algorithms and five machine learning classifiers (logistic regression, support vector machines, Naïve Bayes, K closest neighbor, and random forest). With 85% accuracy, the random forest classifier with top features picked by the MI approach performed best.

Garg *et al*. (2020) proposed an approach to ensure the integrity of online exams by utilizing the Convolutional Neural Network (CNN). Initially, face detection is performed through the Viola-Jones algorithm. Subsequently, CNN is used for face recognition. Finally, the abnormal activities of the examinee are identified based on face recognition and audio input in the background. In another study, (Abisado *et al*., 2018) analyzed a student's gestures to establish behavior consistency with cheating in an online exam. They used a divide and conquer algorithm where feature extraction, such as head poses, is accomplished through HAAR and then classification is performed through HMM. This led to accurately identifying student behavior associated with cheating on online exams. In another study, Chia Yuan (Chuang *et al*., 2017) predicted cheating odds in online exams using the examinee's head poses and time delay. The authors utilized the CLM-GAVAM algorithm for feature extraction from online exam videos and subsequently, prediction is performed through logistic regression. In a similar study, Hu *et al*. (2018) proposed a rule-based reasoning system to detect abnormal behavior via continuous monitoring of the examinee's head poses and mouth state through data gathered using a webcam during an online examination using Adaboost and Haar algorithms.

Tools Evaluation

Various state-of-the-art tools have been reported in an attempt to detect a student's behavior that may lead to cheating. For example, to conduct online examinations securely and reliably on a large scale, Secure Exam Environment (SEE), a proprietary tool, has been proposed by Frankl *et al*. (2019). It ensures the blocking of unauthorized files and information available on internet pages. Al-Hawari *et al*. (2019) proposed a web-based secure and integrated Examination Management System (EMS) developed using Java Enterprise. The system can generate instance forms of the designated examination larger than the maximum capacity of the session and then distribute those forms randomly to students to prevent cheating. Ghizlane *et al*. (2019) proposed a continuous online monitoring system based on face recognition. Several parameters are defined to detect the student's suspicious and abnormal behavior while taking the exam. When these parameters are not respected, the system takes it as abnormal behavior or an attempt to cheat. Various machine learning-based models are suggested to detect these abnormal behaviors. Aisyah and Subekti (2018) presented a continuous authentication system on an Android-based online exam application. The proposed system consists of an authentication and supervision module. Combining the two modules authenticates the examinee and monitors its behavior throughout the exam. In another study, Sabbah (2017) developed an examination authentication system that uses biometrics and user behavior mechanisms for authentication. This authentication system uses facial recognition, fingerprints, and keystroke dynamics. Sabbah (2017) proposed a solution that can detect faults on the users' end, such as internet connectivity termination or shutting down the computer. In another research Kausar *et al*. (2020), authors explored a Secure E-learning System (SES) to share examination-related materials, ensuring protection against various security attacks. They used a fog server and a Session Key Establishment Protocol (SKEP) to set keys for a specified period, such as a class, seminar, or exam. The legitimacy of students is also verified to maintain the trust and authentication level. Biometrics is also reported in Sukmandhani and Sutedja (2019), where the eigenface method is used to authenticate users during an online exam. Opgen-Rhein *et al*. (2018) authors developed FLEXauth, an application for electronic programming exams with machine learning techniques. They discussed the state of the art of author verification, first results, and open research questions that must be addressed for the further development of FLEXauth. Various tools have been summarized in Table 1, along with their availability status. As it may be noticed, these state-of-the-art tools are not freely available and are propriety.

Table 1: A summary of online proctoring solutions in the literature

Sr. #	Approach/ tool name	Availability	Relevant study
1	Secure Exam Environment (SEE)	N-A	Atoum *et al*. (2017)
2	Examination Management System (EMS)	N-A	Garg *et al*. (2020)
3	Continuous online authentication system	N-A	(Abisado *et al*. 2018)
4	Online Exam Proctoring (OEP) system	N-A	Chuang *et al*. (2017))
5	Unified e-examination solution	N-A	Hu *et al*. (2018)
6	Secure E-learning system	N-A	Frankl *et al*. (2019).
7	Prototype online exam app	N-A	Al-Hawari *et al*. (2019)
8	FLEXauth	N-A	Ghizlane *et al*. (2019)

Research Gaps

The reviewed literature highlighted the widespread adoption of E-learning platforms, mainly due to the need for alternative education methods during crises like natural calamities, conflicts, and, more recently, global pandemics. The literature emphasized the security, integrity, and reliability of examinations conducted through these E-learning frameworks and applications, which is crucial for them to be acceptable in educational communities worldwide.

However, there are some limitations to the proposed solutions. Many of these solutions require high-speed internet connectivity. Such solutions cannot be relied upon in developing countries with limited network bandwidth and internet speed. Moreover, many of the solutions presented in the literature are not available freely and come with a financial cost, which can burden educational budgets. They also demand significant computing power due to the inherent complexities of image processing and machine learning algorithms. Finally, these solutions can be difficult to implement in areas with limited access to high-performance computing resources due to the expensive infrastructure required for proper working.

Given these challenges, there is a dire need for a user-friendly solution that can be quickly adopted by the students and academicians of developing countries with minimal wages, slow internet speed, and low bandwidths. Furthermore, this solution should work with low computational power and latency, making it a practical choice for educational institutes in underprivileged areas. A critical research objective in online exam security and e-learning is filling up these gaps in the literature and developing a feasible solution.

Materials and Methods

Our study proposes a solution that can be used in environments with limited internet speed and where the population often struggles to afford costly proctoring systems proposed in the literature. This section discusses the proposed solution in detail. The data acquisition and cleaning are discussed. The feature extraction from the cleaned data using term Frequency (TF) and Inverse Document Frequency (IDF) is discussed. Finally, cheating prediction based on extracted features is discussed.

Data Acquisition and Cleansing

Our approach uses natural language processing to detect online exam cheating. This technique uses a process we summarized as Extract, Load, and Transform (ELT). Data collection is activated when a candidate completes user authentication and loads the exam application. The first step of ELT extraction is gathering data from keystroke logs, applications, and the browser. In particular, keystroke logs focus on the keyboard keys a user presses and releases during the exam. Furthermore, our proctoring system monitors opened applications and web browser use. The focus is on gathering the data generated in the browser's history from the point of starting an exam, including time stamps. The data on the date and time stamp is matched with the exam time to ensure that the data used by the cheating predicting algorithm is generated during the exam. The gathered data is stored separately for each student through ELT's loading component. Each student's directory stores data from keystroke logs, applications, and the browser separately on text files. Loading is followed by transformation, which involves preparing the data for input into this study's proposed algorithm for processing. In our case, transformation has been done via third-party Python libraries that convert the data into arrays for further processing. It involves cleaning the data set as part of loading.

Data cleaning is an essential process that refines the keystroke logs, browser history, and data on opened applications gathered during online assessments. This process involves removing irrelevant information, addressing missing data, standardizing data formats and scales, handling duplicates, filtering out noise and outliers, and ensuring text data is cleaned and preprocessed for Natural Language Processing (NLP) analysis. Data cleaning ensures that the data used for analysis is accurate, consistent, and devoid of noise or inconsistencies. The data from key log strokes, applications, and the browser are separately cleaned. The idea is to prepare the data for processing and enhancing the algorithm's effectiveness. In particular, any data not associated with the alphabetic letters is removed. This includes removing data on special characters, control, or caps logs, in the Fig. 2 code snippet. The data from keystroke logs is cleaned by converting the list of the data files into separate lines where all meaningful words are picked from each line or sentence, as shown in the source code in Fig. 3.

The data gathered on opened applications and the browser was cleaned by eliminating the data gathered about applications irrelevant to this study. The following code snippet in Fig. 4 shows the applications and browsers that were focused on by this study for testing purposes.

```
rep = { 'Space' : " " , 'Escape': "" , 'F2': "" , 'F3': "" , 'F4': "" , 'Delete': "" ,
'PrintScreen': "" ,
        'F5': "" ,  'F6': "", 'F7': "", 'F8': "", 'F9': "" ,'F10': "" ,'F11': "" ,'F12':
"","Pause': "" ,
        'Oemtilde': "",'OemQuestion': "",  'RShiftKey': "",  'LControlKey': "",  'LWin': "",
'LWin': "",
        'Apps': "", 'RControlKey':"",'Left': "", 'Down': "",'Up': "",'Right': "",'Tab':" " ,
'Return':"",
         'LShiftKey':"",  'Oemcomma':"",  'OemPeriod':"" , 'Back':"" , 'OemOpenBrockets':"" ,
'Capital':""}
```

Fig. 2: Source code extract for data cleaning and pre-processing-special characters removal

```
lines = KeyboardLines.split("\n")
Key_lines = [line for line in lines if line.strip() != ""]
```

Fig. 3: Conversion of the data file into separate lines

```
AppHis = ['WhatsApp', 'WINWORD' ,'notepad' , 'Skype']
Browser = [ 'firefox', 'chrome', 'explorer' , 'opera' , 'edge' , 'pale Moon' , 'safari']|
```

Fig. 4: Cleansing of applications and browser data

```
  if(i.find('whatsapp')):
    whatsappcount = 1
 if(i.find('Skype')):
    Skypecount = 1
  if (i.find('notepad')):
    notepadcount = 1
  if (i.find('WINWORD')):
    WINWORDcount = 1
```

Fig. 5: Identification of each application

This process involves identifying specific applications in the corpus and using a counter to keep track of the data, in Fig. 5.

Feature Extraction

Once the data was gathered and cleansed, it underwent a feature extraction process. The feature extraction process is fundamental to natural language processing and machine learning. Several feature extraction techniques are available in text analysis, each tailored to specific analytical objectives. One commonly used approach is the Bag of Words (BoW) model, which represents text as a collection of individual words and their frequencies. While BoW is straightforward and efficient, it cannot distinguish the importance of words within specific documents, making it less suitable for tasks requiring nuanced term importance (Qader *et al*., 2019). Another category of techniques encompasses word embedding, including Word2Vec and GloVe. These methods represent words as continuous vectors, proficient in capturing semantic relationships between words. They prove invaluable in tasks such as sentiment analysis and document clustering. However, they may be less suited when the goal is to assess the importance of terms within individual documents (Singh *et al*., 2022).

We employed Term Frequency (TF) and Inverse Document Frequency (IDF) for feature extraction. TF-IDF is widely recognized for its ability to measure both the local importance of a term within a document (Term Frequency (TF) and the global importance of the term across the entire dataset Inverse Document Frequency (IDF). This dual perspective allows it to effectively capture terms that are significant within a specific context, as well as those that are unique across the entire dataset. Stop words are removed before processing the data with TF-IDF to lower the dimensional space of both text documents (Key and process logs). It should be noted that the TFIDF is a dot product of two vectors (Xiang, 2022). In this case, the first TF-IDF was computed using questions in the experimental question bank vis-à-vis keystroke logs. The idea was to establish a similarity between the data in the keystroke logs and questions in the question bank. A second TF-IDF was also computed between questions in the question bank against the data from the browser history to establish similarities. The browser history focused on the searched data compared to questions in the question bank using TF-IDF. The code snippets in Figs. 6-7 show the similarity computation using keystroke logs and browser history.

Computing Cheating Prediction

The data from the previous section is used to calculate the average similarity per question to predict cheating. Several statistical techniques are available for this purpose. One common statistical technique that can be used is logistic regression. Although logistic regression works well for binary classification, it may fail to capture the complex relationship between many data sources (Zou *et al*., 2019).

Decision tree analysis is another frequently considered method since it can reveal complex patterns in the data. However, it might not be probabilistic enough to produce accurate cheating probability and it might have trouble with the weighted integration of many data sources. In addition, methods such as neural networks are occasionally investigated. However, their complexity level and the resources they require may make them less suitable for this particular application (Charbuty and bdulazeez, 2021).

We used the normalized exponential function or softmax activation function to compute probabilities for each vector. The decision to utilize this approach is driven by the requirement for a standardized and rapidly increasing function that can efficiently transform raw similarity vectors into probability distributions. The inherent characteristics of the softmax function make it highly suitable for this task. This feature guarantees that the total probabilities of the output sum up to 1, which is consistent with the probabilistic framework used for predicting instances of cheating (Mercioni and Holban, 2020). Soft Max's ability to produce a clear and understandable output that shows the probability of cheating for each question is one of its main advantages. This degree of detail is necessary for educators and organizations to pinpoint problem areas in an evaluation and implement focused interventions, including

reviewing certain questions or student responses. Figure 8 demonstrates how the softmax activation function was fed with vectors or similarities to compute the cheating probabilities.

The average similarities and probability results are computed for the final cheating prediction. Since students have different preferences for sources, the internet is regarded as a popular source. It was assumed that students' preferences for one technology in cheating differ from the other. For instance, students are more likely to cheat through the internet browser, where they can quickly search for answers. Hence, computations on the data from keystroke logs, browsers, and applications were given different weights. Browser history similarity results (BroSim) were weighted 60%, while keystroke logs similarity results (KeySim) were weighted 40%, Fig. 9.

After weighting, the results on browser history similarity and keystroke log similarities were expressed as 80%. While results from application similarity were defined as 20%. These were added together to give the overall cheating prediction of a student.

This way, the model takes into account all the different factors that affect a student's decision to cheat, which is especially important in today's digital world. The goal of this multifaceted method is to provide educators and institutions with a more accurate way to spot possible cases of cheating in online environments.

```
Key_similarities=np.dot(tfidf_vectors_key,tfidf_vectors_key.T).toarray()
```

Fig. 6: Identification of key similarity

```
Search_similarities=np.dot(tfidf_vectors_ques,tfidf_vectors_ques.T).toarray()
```

Fig. 7: Identification of search similarity

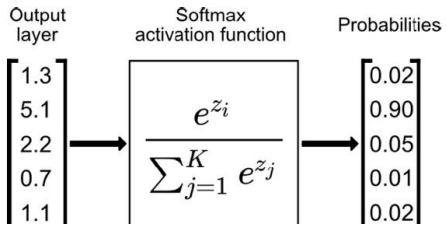

Fig. 8: Computation of the cheating probabilities

$$BroSim = \frac{Sum(Avg\_Search\_Similarity)}{Len\ (Avg\_Search\_Similarity)} * 0.6$$

$$KeySim = \frac{Sum(Avg\_key\_Similarity)}{Len\ (Avg\_key\_Similarity)} * 0.4$$

Fig. 9: Calculation of weighted similarity

Implementation

Within academic assessments, the emergence of online examination proctoring systems has served a crucial role in safeguarding the authenticity and confidentiality of remotely administered exams. This section provides a comprehensive analysis of the implementation details of our online exam proctoring solution, integrating both client-side and server-side components to ensure a comprehensive and secure examination environment.

The detailed architecture of the proposed solution is given in Fig. 10. The first part of the figure represents the client side (examinee or student writing the exam), while the other part represents the server side that could be set up in the cloud.

Client Interfaces

The client-side architecture of our online exam proctoring tool is based on the integration of C# and .NET, which offers a reliable foundation for constructing a desktop program specifically designed for academic evaluations. This framework ensures a secure environment for the development process. The user authentication interface prioritizes user-friendliness and security using the "system windows forms" library. To ensure secure access, students are prompted to provide their name, roll number, login, and password Fig. 11.

The "Cloudinary" API facilitates communication between the cloud server and the user, guaranteeing a safe and easy transfer of exam-related data. The exam interface improves exam creation and resource retrieval flexibility using the "Cloudinary DotNet" and "Excel data reader" libraries. In order to facilitate a controlled assessment environment, real-time monitoring is made possible by the integration of a "timer" library, which graphically displays the remaining exam time Fig. 12. Additional measures to preserve exam integrity include background keystroke logging and process monitoring, which are made possible by the "get time stamp" function.

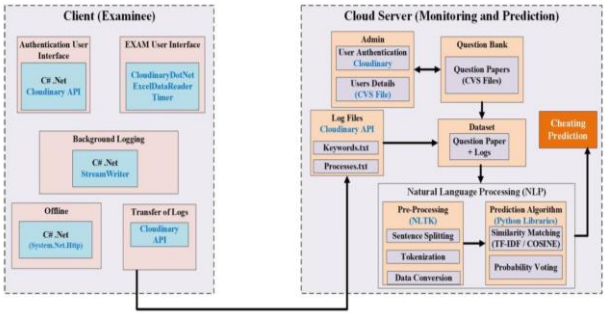

Fig. 10: The architecture of the proposed solution showing the setup on the client and server-side

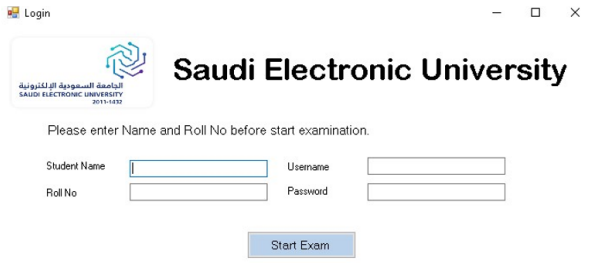

Fig. 11: Login screen

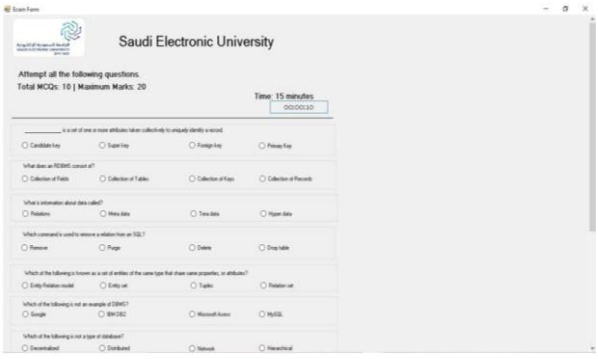

Fig. 12: Exam screen displaying the remaining time

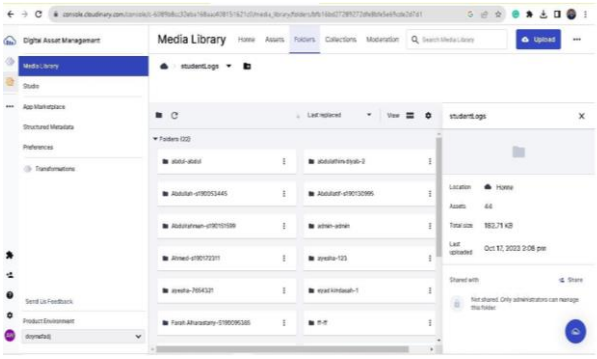

Fig. 13: Cloud folder student submission

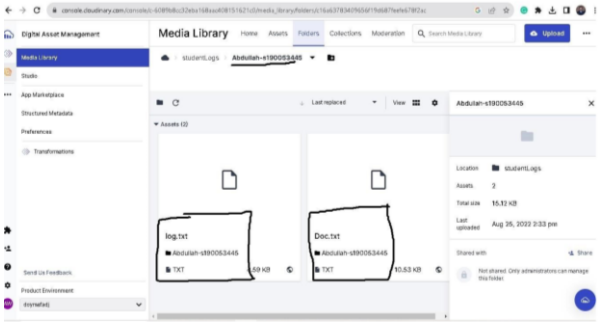

Fig. 14: Log files for each student

Considering the possibility of internet disruptions, the client side of the system has been developed to enable offline monitoring. This is achieved by employing the ".net library system net http," which ensures that the examination continues and monitoring can be carried out even when connectivity is troubling.

Server Interfaces

The server-side design is enhanced with an Application Programming Interface (API) implemented on the "Cloudinary" platform. This integration establishes a secure connection between the client and server, facilitating efficient data interchange. Administrators possess significant authority since they can input examination inquiries and oversee multiple facets of the proctoring method to ensure efficient examination administration.

The server side utilizes sophisticated Natural Language Processing (NLP) techniques by using the "NLTK Stop words" module and the "NUMPY" TF-IDF vectorizer to detect incidents of cheating. The method of data pre-processing encompasses the elimination of frequently occurring stop words and the conversion of textual data into numerical representations.

This enables the system to effectively discover significant patterns and similarities within students' responses, hence enhancing the efficacy of the cheating detection mechanism. The "softmax" library is incorporated into the server side to improve the cheating detection process. By converting average similarity results into probabilities, this statistical technique provides a more sophisticated evaluation of possible copying or cheating. Soft max's implementation improves the system's capacity to identify intricate patterns, leading to a more precise assessment.

The data is carefully organized and a separate dataset is used to store students' answer papers safely. Each student has a folder in the "student logs" section that holds two important files: "Log.txt" and "Doc.txt" in Figs. 13-14. These files keep track of the keys pressed and the programs opened during the assessment. This gives administrators detailed information they can use to check the authenticity of the exam environment.

Results

The effectiveness of cheating detection by the proposed proctoring solution was evaluated through an experiment at a university in Saudi Arabia. This is a common trend in literature: The proposed solutions are evaluated through data gathered from an experiment. Sixteen students from a bachelor of science degree

program (software engineering) were selected for this study's experiment. The students were divided into two equal groups of 8 students. The groups were given an online Software engineering and databases exam, respectively. This exam was based on Multiple Choice Questions (MCQs). Each exam had 30 MCQs and the duration of the exams was 20 min. Students in group 1 were asked to behave as they would in an exam. In contrast, students in group 2 were allowed to cheat by searching for answers on the internet and other applications they considered helpful. This exam was conducted with our proposed proctoring solution, monitoring students for cheating behavior. The results from this experiment are summarized in Table 2. The first column represents each student who took part in the experiment. Each student is uniquely identified using a pseudo name: The word student followed by a number. The students with cheating intent are marked "Yes" in the second column. The similarity index of key logs and browser history concerning the question paper is given in the third column. The cheating likelihood based on the application history (i.e., Skype, WhatsApp, PDF, notepad PowerPoint, Word, Excel) during the exam is given in the fourth column. The overall value of cheating likelihood (similarity index + application history) is shown in the fifth column. Finally, based on the total and given threshold values, the actual prediction like "fair" or "cheating" is provided in the last column. Students with an overall cheating prediction greater than 12 (column five) were adjudged to have cheated, while those with a score below 12 were considered to have honestly (fair) attempted the exam. It is important to note that the threshold value for cheating prediction is highly important. A low threshold value may lead to frequent false positives, i.e., fair students may be predicted as cheaters. On the other hand, the high threshold values may lead to missing the cheating cases. However, this study's set threshold value (12) can be adjusted per the real exam environment to achieve optimum results.

It can be seen from Table 2 that students 2, 5, 7, and most students from group 2 are tempted to cheat during the exam. Similarity predictions show these students cheated using the browser, suggesting that answers were searched online. High similarity predictions on Keystroke logs affirm the possibility of searching for information online. Therefore, these students' similarity index (key logs and browser history) is relatively high compared to others. It can also be seen from Table 2 that all students in Group 1 have some minor values associated with application history. This is because only some applications like Notepad are frequently used so that students may open such applications during exams without cheating intentions. Finally, the proposed framework successfully predicted all cheating cases, as given in the last column of Table 2.

Table 2: Group-wise results of the solution

Reg. #	Cheating intent	Similarity (80%) index		App History (20%)	Total (80+20%)	Actual prediction (as per threshold limit)
		Key logs	Browser			
Group 1 results						
Student 01	No	0.0001	0.0	0.002	0.0021	Fair
Student 02	Yes	8.2000	5.5	0.003	13.0703	Cheating
Student 03	No	0.0003	0.0	0.004	0.0043	Fair
Student 04	No	0.0000	0.0	0.002	0.0002	Fair
Student 05	Yes	10.5000	7.2	0.001	17.0701	Cheating
Student 06	No	0.0000	0.0	0.001	0.0010	Fair
Student 07	Yes	9.7000	8.8	0.006	18.5060	Cheating
Student 08	No	0.0000	0.0	0.003	0.0030	Fair
Group 2 results						
Student 09	Yes	7.3000	5.0	1.002	13.5000	Cheating
Student 10	Yes	8.2000	6.1	1.009	16.2000	Cheating
Student 11	Yes	7.8000	5.6	2.004	15.8000	Cheating
Student 12	Yes	9.5000	7.5	1.000	18.0000	Cheating
Student 13	Yes	13.5000	10.5	0.009	24.9000	Cheating
Student 14	Yes	10.2000	8.2	0.004	18.8000	Cheating
Student 15	Yes	5.7000	4.2	1.001	11.0000	Fair
Student 16	Yes	8.5000	6.2	2.003	17.0000	Cheating

Discussion

Global events and technological improvements have led to a rise in online learning, which has increased the demand for reliable exam-proctoring solutions. This is particularly important in the context of academic institutions and certification bodies looking to preserve the integrity of their assessments. Many proctoring techniques have been proposed in response to this requirement; some depend on sophisticated hardware and strong internet connections. However, their cost is the primary challenge to the widespread acceptance of these advanced proctoring methods. Students frequently struggle with a lack of funding in many places, especially in developing countries. Acquiring strong PCs and having access to fast internet might be major challenges. This is where the novel approach used in this study comes into work.

This study focused on proposing an online proctoring solution that does not rely on expensive hardware and high internet speed. We argue that everyone can't afford high-end proctoring solutions proposed in the literature that require high internet speed to support visual and audio data to predict cheating. The population in underdeveloped countries can barely afford entry-level computers and high-speed internet connectivity. The study presents a method for proctoring online tests that is both cost-effective and resource-efficient, utilizing the capabilities of Natural Language Processing (NLP). Natural Language Processing (NLP), a subfield of artificial intelligence, facilitates the analysis of written and typed textual data, hence enabling the identification of unusual patterns or evidence of cheating.

The internet is one of the primary sources of information that students use to give answers to questions and Natural Language Processing (NLP) can analyze diverse data sources to assess the probability of engaging in cheating. This study successfully showed that data from the browser history, keystroke logs, and applications can be used to predict cheating during the exam using NLP. The examination of browsing history has the potential to unveil if a student has accessed external websites during the examination, indicating the possibility of searching for answers or obtaining unauthorized materials. The analysis of keystroke logs can offer valuable insights into the pace and frequency of typing, enabling the detection of irregularities that could potentially suggest collusion or unlawful aid. Similarly, monitoring the utilization of particular applications can aid in identifying behavioral patterns that depart from the established standard in the context of an examination.

This solution does not require high-end hardware such as a camera for capturing the videos, audio and a high internet speed, making this highly affordable. In addition, while this study's proposed solution could be used independently, it can also be used with other proctoring solutions presented in the literature to reduce internet costs and other computing resources. In this case, this study's low internet consumption solution can be used to monitor a candidate's behavior throughout the exam. However, when the calculated cheating scores reach a set threshold, other verification security measures can be activated to gather evidence. Such evidence may be video or audio-based. This is a common in-person exam where the proctor would visit a suspected cheating candidate to gather evidence that will be used for further disciplinary action. Furthermore, the proposed solution may be incorporated into learning management systems to collect student behavior data when attempting informative assessments based on MCQs. For example, the proposed solution can be used in MCQs-based quizzes where the instructor may gather data on students' use of different sources of information when attempting the assessments. Students with limited domain knowledge will likely guess answers in an evaluation. Hence, students' willingness to search for answers online instead of guessing may help the instructor better understand students' commitment to the subject.

In short, the paper introduces an NLP-based proctoring system that effectively tackles the significant challenges related to cost, accessibility, and efficacy in online exam proctoring. The proposed strategy presents a cost-effective and adaptable solution and aligns with the primary goals of improving the credibility of online assessments and encouraging fair and accessible educational practices.

Limitations

This study's proposed solution has some limitations:

- Students can use a second device to search for information. However, incorporating this technique with other multimodal authentication mechanisms may enhance exam security
- NLP-based tools, like the one proposed, can produce false positives, flagging innocent behavior as cheating. This can lead to unnecessary scrutiny of students who have not engaged in any misconduct
- Collecting and analyzing keystroke logs, browser history, and application data raises significant privacy concerns. Students may be uncomfortable with their online activity being monitored and recorded, potentially leading to ethical issues
- The proposed solution is most appropriate when proctoring an MCQs-based exam. However, keystroke logs may lead to false positives in an essay-based exam

Conclusion

This study demonstrated that NLP can predict cheating using keystroke logs, browser history, and data on opened applications during an online assessment. The data extracted from these resources was cleaned to remove unnecessary information. The cleaned data was then compared with the questions in questionnaires through TF-IDF to predict cheating. The proposed solution was evaluated in an experiment with sixteen undergraduate software engineering students at a university in Saudi Arabia. The students were divided into two groups of eight students. The groups were given software engineering and database exams, respectively and only one group was allowed to cheat during the exam. Each exam was 20 min long and had 30 MCQs. The findings revealed that the proposed proctoring tool effectively predicts cheating during the exams. The results bring the hope of attempting proctored online exams to communities that cannot afford high-speed internet connectivity and pricey powerful computers. This study addresses the continued digital divide between the haves and the haves not because of affordability and access to a high-speed internet connection associated with most exam proctoring algorithms in the literature. This study will pave the way for an interest in investigating exam cheating solutions that could be more resourceful. However, more research is needed to develop exam proctoring solutions that do not require high internet speed and expensive hardware components for enabling biometrics or capturing visuals and audio.

Acknowledgment

Author wants to pay special gratitude to all the individuals who directly or indirectly involved in this study.

Funding Information

The authors confirmed that there is no funding agency exists in this study.

Ethics

This manuscript is original and contains unpublished material. There is no conflict of interest.

References

Abisado, M. B., Gerardo, B. D., Vea, L. A., & Medina, R. P. (2018, July). Towards academic affect modeling through experimental hybrid gesture recognition algorithm. In *Proceedings of the 2018 International Conference on Data Science and Information Technology*, (48-52). https://doi.org/10.1145/3239283.3239305

Abisado, M. B., Rodriguez, R. L., Arias Jr, A. R. V., Isip, C. M. M., Bungay, J. D. D., Cipriano, J. M. V., & Vea, L. A. (2019). Modeling filipino academic affect during online examination using machine learning. *In Proceedings of the 20th Annual SIG Conference on Information Technology Education*. 167-167. https://doi.org/10.1145/3349266.3351387

Aisyah, S., & Subekti, L. B. (2018, October). Development of continuous authentication system on android-based online exam application. In *2018 International Conference on Information Technology Systems and Innovation (ICITSI)*, (171-176). IEEE. https://doi.org/10.1109/ICITSI.2018.8695954

Al-Hawari, F., Alshawabkeh, M., Althawbih, H., & Abu Nawas, O. (2019). Integrated and secure web-based examination management system. *Computer Applications in Engineering Education*, *27*(4), 994-1014. https://doi.org/10.1002/cae.9

Andersen, K., Thorsteinsson, S. E., Thorbergsson, H., & Gudmundsson, K. S. (2020, April). Adapting engineering examinations from paper to online. *In 2020 IEEE Global Engineering Education Conference (EDUCON)*, (pp. 1891-1895). IEEE. https://doi.org/10.1109/EDUCON45650.2020.9125273

Atoum, Y., Chen, L., Liu, A. X., Hsu, S. D., & Liu, X. (2017). Automated online exam proctoring. *IEEE Transactions on Multimedia*, *19*(7), 1609-1624. https://doi.org/10.1109/TMM.2017.2656064

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20-28. https://doi.org/10.38094/jastt20165

Chuang, C. Y., Craig, S. D., & Femiani, J. (2017). Detecting probable cheating during online assessments based on time delay and head pose. *Higher Education Research and Development*, *36*(6), 1123-1137. https://doi.org/10.1080/07294360.2017.1303456

Fan, Z., Xu, J., Liu, W., & Cheng, W. (2016). Gesture based misbehavior detection in online examination. *In 2016 11th International Conference on Computer Science and Education (ICCSE)*, (234-238). IEEE. https://doi.org/10.1109/ICCSE.2016.7581586

Frankl, G., Napetschnig, S., & Schartner, P. (2019). Pathways to successful online testing: Exams with the "Secure Exam Environment" (SEE). In *Computer Supported Education: 10th International Conference, CSEDU 2018, Funchal, Madeira, Portugal, March 15-17, 2018, Revised Selected Papers 10*, (231-250). Springer International Publishing. https://doi.org/10.1007/978-3-030-21151-6_12

Garg, K., Verma, K., Patidar, K., & Tejra, N. (2020, May). Convolutional neural network based virtual exam controller. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, (895-899). IEEE. https://doi.org/10.1109/ICICCS48265.2020.9120966

Garg, M., & Goel, A. (2023). Preserving integrity in online assessment using feature engineering and machine learning. *Expert Systems with Applications*, *225*, 120111. https://doi.org/10.1016/j.eswa.2023.120111

Ghizlane, M., Hicham, B., & Reda, F. H. (2019, December). A new model of automatic and continuous online exam monitoring. In *2019 International Conference on Systems of Collaboration Big Data, Internet of Things and Security (SysCoBIoTS)*, (pp. 1-5). IEEE. https://doi.org/10.1109/SysCoBIoTS48768.2019.9028027

Gopane, S., & Kotecha, R. (2022). Enhancing monitoring in online exams using artificial intelligence. *In Proceedings of International Conference on Data Science and Applications: ICDSA 2021*, *2* (pp. 183-193). Springer Singapore. https://doi.org/10.1007/978-981-16-5348-3_14

Gudiño Paredes, S., Jasso Peña, F. D. J., & de La Fuente Alcazar, J. M. (2021). Remote proctored exams: Integrity assurance in online education. *Distance Education*, *42*(2), 200-218. https://doi.org/10.1080/01587919.2021.1910495

Hussein, M. J., Yusuf, J., Deb, A. S., Fong, L., & Naidu, S. (2020). An evaluation of online proctoring tools. *Open Praxis*, *12*(4), 509-525. https://search.informit.org/doi/abs/10.3316/informit.620366163696963

Hu., Jia, X., & Fu, Y. (2018, August). Research on abnormal behavior detection of online examination based on image information. *In 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, *2*(88-91). IEEE. https://doi.org/10.1109/IHMSC.2018.10127

Kaddoura, S., & Gumaei, A. (2022). Towards effective and efficient online exam systems using deep learning-based cheating detection approach. *Intelligent Systems with Applications*, *16*, 200153. https://doi.org/10.1016/j.iswa.2022.200153

Kasinathan, V., Yan, C. E., Mustapha, A., Hameed, V. A., Ching, T. H., & Thiruchelvam, V. (2022). ProctorEx: An Automated Online Exam Proctoring System. *Mathematical Statistician and Engineering Applications*, *71*(3s2), 876-889. https://philstat.org/special_issue/index.php/MSEA/article/view/320

Kausar, S., Huahu, X., Ullah, A., Wenhao, Z., & Shabir, M. Y. (2020). Fog-assisted secure data exchange for examination and testing in e-learning system. *Mobile Networks and Applications*, 1-17. https://doi.org/10.1007/s11036-019-01429-x

McCoy, C., Yu, A., & Ramazanova, S. (2015). An author co-citation analysis: Examining the intellectual structure of e-learning from 1981-2014. *Proceedings of the Association for Information Science and Technology*, *52*(1), 1-3. https://doi.org/10.1002/pra2.2015.145052010090

Mercioni, M. A., & Holban, S. (2020, May). The most used activation functions: Classic versus current. *In 2020 International Conference on Development and Application Systems (DAS)*, (pp. 141-145). IEEE. https://doi.org/10.1109/DAS49615.2020.9108942

Mohammed, Hussein M. & Qutaiba I. Ali. (2023). "Cheating Detection in E-exams System Using EEG Signals." *International Conference on Scientific and InnovativeStudies*, *1*. No. 1. https://doi.org/10.59287/icsis.601

Muzaffar, A. W., Tahir, M., Anwar, M. W., Chaudry, Q., Mir, S. R., & Rasheed, Y. (2021). A systematic review of online exams solutions in e-learning: Techniques, tools and global adoption. *IEEE Access*, *9*, 32689-32712. https://doi.org/10.1109/ACCESS.2021.3060192

Ngqondi, T., Maoneke, P. B., & Mauwa, H. (2021). A secure online exams conceptual framework for South African universities. *Social Sciences and Humanities Open*, *3*(1), 100132. https://doi.org/10.1016/j.ssaho.2021.100132

Opgen-Rhein, J., Küppers, B., & Schroeder, U. (2018, November). An application to discover cheating in digital exams. *In Proceedings of the 18th Koli Calling International Conference on Computing Education Research*, (pp. 1-5). https://doi.org/10.1145/3279720.3279740

Palvia, S., Aeron, P., Gupta, P., Mahapatra, D., Parida, R., Rosner, R., & Sindhi, S. (2018). Online education: Worldwide status, challenges, trends and implications. *Journal of Global Information Technology Management*, *21*(4), 233-241. https://doi.org/10.1080/1097198X.2018.1542262

Qader, W. A., Ameen, M. M., & Ahmed, B. I. (2019, June). An overview of bag of words; importance, implementation, applications and challenges. *In 2019 International Engineering Conference (IEC)*, (pp. 200-204). IEEE. https://doi.org/10.1109/IEC47844.2019.8950616

Sabbah, Y. W. (2017). Security of online examinations. *Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications*, 157-200. https://doi.org/10.1007/978-3-319-59439-2_6

Sarmiento, M. J. F., Lopez, V. A., Altoveros, L. A. D., Tiosan, J. C. B., Tabo, R. M. R., & Manangat, R. D. M. (2023, May). Tracking Input Devices to Detect Cheating Using Machine Learning Techniques. *In 2023 8th International Conference on Business and Industrial Research (ICBIR)*, (519-524), IEEE. https://doi.org/10.1109/ICBIR57571.2023.10147616

Singh, K. N., Devi, S. D., Devi, H. M., & Mahanta, A. K. (2022). A novel approach for dimension reduction using word embedding: An enhanced text classification approach. *International Journal of Information Management Data Insights*, *2*(1), 100061. https://doi.org/10.1016/j.jjimei.2022.100061

Slusky, L. (2020). Cybersecurity of online proctoring systems. *Journal of International Technology and Information Management*, *29*(1), 56-83. https://doi.org/10.58729/1941-6679.1445

Sukmandhani, A. A., & Sutedja, I. (2019, August). Face recognition method for online exams. *In 2019 International Conference on Information Management and Technology (ICIMTech)*, (*1*, pp. 175-179). IEEE. https://doi.org/10.1109/ICIMTech.2019.8843831

UNDESA. (2022). *United Nations E-Government Survey 2022: The Future of Digital Government*. UN. https://doi.org/10.18356/9789210019446

Xiang, L. (2022). Application of an improved TF-IDF method in literary text classification. *Advances in Multimedia*, *2022*. https://doi.org/10.1155/2022/9285324

Zou, X., Hu, Y., Tian, Z., & Shen, K. (2019, October). Logistic regression model optimization and case analysis. *In 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)* (pp. 135-139). IEEE. https://doi.org/10.1109/ICCSNT47585.2019.8962457