AN INTELLIGENT SYSTEM TO AUTOMATE THE DETECTION OF ONLINE CHEATING ACTIVITIES USING AI AND CONTEXT AWARE TECHNIQUES

Qinyuhan Zhao[1], Mingze Gao[2] and Yu Sun[3]

[1]Pacific Academy Irvine, 4947 Alton Pkwy, Irvine, CA 92604
[2]University of California, Irvine, 204 Aldrich Hall. Irvine, CA 92697
[3]California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

In the environment of online courses and online exams, cheating in online courses is prevalent [1]. To better ensure fairness in exams, schools and educational institutions need to use technology to detect and deter cheating [2]. Starting from practical application, this paper discusses 3 different methods to detect cheating behavior, and proposes a new way. for online exam supervision.

KEYWORDS

Machine Learning, Audio Detection, Context-Aware.

1. INTRODUCTION

At the end of 2019 a disease called covid-19 is sweeping across the world, and it has soon become a global disaster [3]. After that people on earth entered the age of pandemic, where people stay at their houses to prevent the spread of the disease. Shortly after people entered the pandemic remote learning came into people's sight and it has changed people's understanding of online learning drastically [4]. We are amazed at how advanced our technologies have become so that it allows all of us to meet online to study even though at our own house. Huge progress has been made by humans, but still, there are a lot of flaws in this system.

As soon as the pandemic entered the end of the first year I noticed a serious problem, a problem that's gonna put the whole human race on a string between life and death. I found out that many of my friends and classmates that I know are all cheating on their quizzes and tests to achieve better grades, and this perfectly demonstrates the flaws of online learning [5]. Different from in-person teachers were able to check on the environment around students in real-time, but when it comes to online quizzes and tests teachers are not able to see the environment around the students to help them demonstrate academic honesty. After I realized this has become a serious issue out of curiosity I did a lot of research online and the result shows the rate of academic dishonesty that occurred during the pandemic has increased significantly in comparison to in-person studies. This "blind spot" from the teachers gives a lot of students chances to cheat during their exams, and to put this in perspective soon there will be surgeons that cheat on their biology class in high school performing surgeries for us. I realize this going to become a national crisis that must be stopped so I decided to do something about it. I started to look into ways to prevent. online cheating.

David C. Wyld et al. (Eds): SIP, CRBL, WiMNeT, BIOM, EDUPT, NLDM, SOFTFM, COMIT - 2022
pp. 67-74, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121707

There are a lot of apps or websites that offer ways to prevent cheating but none of them was ideal enough to make sure the student doesn't get a chance to cheat [6]. Ever since that, I made up my mind to create a program of my own to prevent the issue of online academic dishonesty.

There are actually a lot of methods and applications on the internet to prevent academic dishonesty/ online cheating. However, many of the anti-cheating techniques and systems that have been developed are not working effectively. An obvious example is College board uses a lockdown browser as their method to make sure the student can't cheat on their test, but in reality, the students that are taking the test might have their book opened and possibly another device on the side helping them search for formulas they need to use for that certain question. By just using a lockdown browser students might also just ask for their friend's or tutor's help during the exam. The lockdown browser sounds prominent with accountability but in reality, there's no guarantee that the student doesn't cheat because there are too many flaws in this method that allows the student to have the chance to cheat on their exam. Another method that I usually see online is that during the exam I will take a picture of the test taker to see if they are the actual person that's taking the exam. I think this is a smart idea but it's not ideal. The benefit is that the exam will know whether the student is actually taking the exam or other people that are helping the student taking the exam, but on the other hand, the student might communicate with other people during the exam through another device to establish communication with their friends or tutor to help them during the exam. The intention of this system and methods are good, but they are not working ideally as they are expected to be because there are flaws and imperfections within them.

My method is to utilize a sound monitor and mouse movement tracking technology in our system to ensure cheating is impossible during the exam [7]. The current existing method is to lockdown browsers but students might cheat by getting outside help at their house. My sound monitoring system ensured that students cannot talk to other people during the exam. If they get any outside help the system will capture the voice and the student will be determined as cheating.

We set up three experimental groups and the first one is the control group by not using any anti-cheating application during the exam, the second group uses a lockdown browser during the exam, and the third group uses the application I developed. We first placed the three candidates in three separate empty rooms, and then we gave them three exact same exams and another computer. The first candidate was able to cheat easily because it doesn't have any application or any restriction on him. The second candidate had a locked-down browser on the test he had so he used another computer to call his friend for help and was able to easily pass the exam too. The third candidate used my application and was trying to do the same as what the second candidate was trying to do with the other laptop but he was determined to be cheating because the sound system recognized the candidate was talking to someone else. This shows my application does work.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section5. Finally, Section 6 gives the conclusion remarks, as well as pointing out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follow.

My largest challenge is to learn the programming language python because without it I can't even start my project [8]. At first, I didn't know what to do, so I asked to do some research online. I found out that there are many education centers that can teach python. After I checked most of them I decided to go to coding minds and study python there. At first, I was struggling with the coding process and I almost gave up, but when I think about how my program might benefit the whole society. I kept learning python until I was capable of doing the project, and eventually, I accomplished my goal.

3. SOLUTION

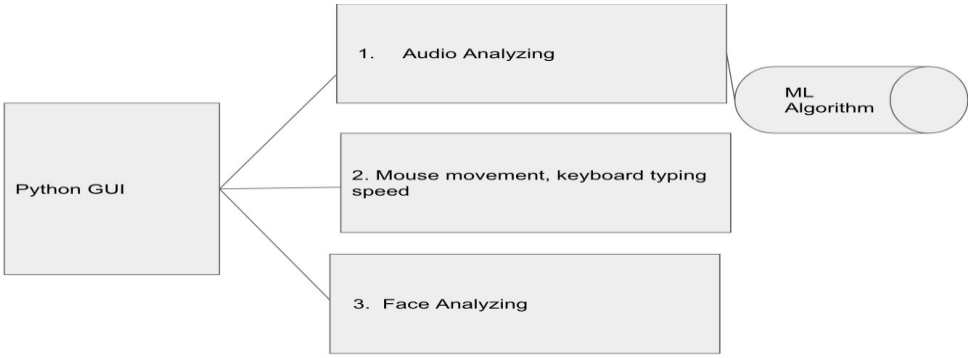

Figure 1. Overview of the solution

The cheating detection system is accomplished by using the language of Python that the system supports based on three main features which are audio analyzing, mouse movement with keyboard typing speed and face analyzing (Figure 1). The current audio analyzing system takes steps to calculate the user's sound wave amplitude that compares with the default value detecting any conversation or discussion behavior during the test. In the future, the system is planning to collect and train the data from users at the same time, where the machine learning algorithms will be implemented in order to predict and analyze the sound from users [9]. By doing this, machine learning will capture and learn the relationships between data providing the result of the detection. The process of checking the mouse movement with keyboard typing speed also plays an important role to catch any abnormal activities during the exams or quizzes. For example, if the mouse moves too often or the keyboard types too fast, those frequent signs will cause an alert and warning for cheating detection. In the meantime, the system will activate the face capture that monitors all the sensitive facial movements such as mouth movement, eye movement, head movement, etc. By combining all three results from the detection features, the system will make a decent result based on the user's activities under the detection. The three detection features have minimized the deviation of the result. The system instructions are easy to follow, and users will download the software on their laptops or desktops. Once the software is successfully downloaded and opened, tests will be provided by the instructor. The detector will be activated automatically once the exam is started. When users are finished taking the exams, students are required to click the "Submit Exam" button, then the system will provide the relevant results to the instructor.

Two algorithms have been implemented into the feature of mouse detection where the algorithm will check two factors, boundaries and rapid movements. When the mouse has either vertically or horizontally moved outside of the testing window, warnings will appear. In addition, if the mouse has moved too fast that exceeds a certain range of speed, this behavior will be detected as a rapid movement which causes warnings as well. The cheating will be detected based on 2 or more combinations of warnings. For example, if a student tries to open a browser during the exam, he

or she quickly moves the mouse outside of the testing window. This kind of behavior will cause at least two warnings that the mouse goes over the boundary and the moving speed is exceeded. The warnings will cause the cheating detection to be activated, and the system will immediately stop the exam, generating the report to the instructor at the same time [10]. As shown in Figure 2, warning detection are implemented based on different factors. The algorithm uses a number of four warnings to determine both mouse out-of-bound and rapid movements. If the warning has been detected, it will be set to TRUE. The 'warning1' will be TRUE if the mouse moves too fast to the left or right, and 'warning2' will focus on the speed of the mouse moving up and down. In the meantime, the 'warning3' will be set to TRUE if the mouse moved out of the bond horizontally, while 'warning4' checks whether the mouse moved out of the bond vertically. If 2 or more warnings are being set to TRUE, the cheating will be detected. For example, once warning1 and warning3 are set to TRUE, the algorithm will shut down the exam, and the student will be prohibited from taking the test. In addition, if warning1, warning2, and warning3 have been detected, the exam will be ended as well, as long as there are at least two warnings set to TRUE.

```
def detect_warnings(self, x, y):
  warning1 = bool
  warning2 = bool
  warning3 = bool
  warning4 = bool

  if self.x != 0 and self.y != 0:
    # Rapid Movements
    if ((self.x - x) > 225) or ((x - self.x) > 225):
      self.mouse_movements.append("Warning-1")
      warning1 = True
    if ((self.y - y) > 225) or ((y - self.y) > 225):
      warning2 = True
      self.mouse_movements.append("Warning-2")
    # Out-of-bounds
    if x >= 448 or x <= 4:
      warning3 = True
      self.mouse_movements.append("Warning-3")
    if y >= 411 or y <= 4:
      warning4 = True
      self.mouse_movements.append("Warning-4")
  self.x, self.y = (x, y)
```

Figure 2. Mouse Movement Detection Code

The sound detection feature will make sure that the environment around the student is quiet enough, so that it detects whether the student is talking to someone else during the exam or not. The sound detection will check the amplitude of the sound waves instead of reading the words or sentences from the user. The system checks sound waves every 100ms. If the current sound wave. is greater than the initialized sound amplitude, the algorithm will identify that the student has cheated during the exam because of talking. Based on Figure 3, the proper measurements and global requirements for audio recording have been initialized for sound identification. For example, if the collected sound amplitude from the student is greater than the initial value of 'VOICE_THRESHOLD' which means the noise or sound is louder than normal. Since the value of 'VOICE_THRESHOLD' has been carefully calculated, it avoids the small noise or sounds during the exam such as typing, sketching, clicking, etc.

```
VOICE_THRESHOLD = 0.28
FORMAT = pyaudio.paInt16
SHORT_NORMALIZE = (1.0/32768.0)
CHANNELS = 1
RATE = 44100
INPUT_BLOCK_TIME = 0.05
INPUT_FRAMES_PER_BLOCK = int(RATE*INPUT_BLOCK_TIME)
```

Figure 3. Initialized Measurements for Sound Detection

By looking at Figure 4, it shows the main process of sound detection where the algorithm will first check to select the appropriate input device [11]. If there is no preferred input, the system will set the input device as default. Once the input has been set, the function of 'open_mic_stream' will set all the needed requirements such as format, channels, rate, etc. When the input device is ready and the microphone stream is opened, the sound detector will check the amplitude every round to compare with the 'VOICE_THRESHOLD' that has been initialized. The algorithm will always return TRUE if the sound exceeds the maximum amplitude. If the current amplitude does not encounter any problem, the next sound amplitude will be checked after 100ms.

```
def find_input_device(self):
    device_index = None
    for i in range( self.pa.get_device_count() ):
        devinfo = self.pa.get_device_info_by_index(i)
        print( "Device %d: %s"%(i,devinfo["name"]) )

        for keyword in ["mic","input"]:
            if keyword in devinfo["name"].lower():
                print( "Found an input: device %d - %s"%(i,devinfo["name"]) )
                device_index = i
                return device_index
    if device_index == None:
        print( "No preferred input found; using default input device." )
    return device_index

def open_mic_stream(self):
    device_index = self.find_input_device()
    print(device_index)
    stream = self.pa.open(  format = FORMAT,
                            channels = CHANNELS,
                            rate = RATE,
                            input = True,
                            input_device_index = device_index,
                            frames_per_buffer = INPUT_FRAMES_PER_BLOCK)
    return stream

def detector(self):
    block = self.stream.read(INPUT_FRAMES_PER_BLOCK, exception_on_overflow = False)
    amplitude = get_rms(block)
    if amplitude > VOICE_THRESHOLD:
        return True
    else:
        return False
```

Figure 4. Sound Amplitude Detection Code

The third detection feature is face movement analyzing which is still in progress [12]. The plan is broken into two sub-parts. The first part will be setting the dimensions of the camera to monitor the student's face within specific boundaries by considering enough space for normal movements. The second part is designing a face movement capture system that catches all the sensitive facial movements. For example, if the eyes are looking outside of the screen, the system will identify it as a warning. If the eyes are looking outside of the screen for longer than 5 seconds, then the algorithm will detect this behavior as cheating. Similar algorithms will apply to head movements and mouse movements as well, which checks the degree of head rotation and frequency of mouth movement during the exam.

4. EXPERIMENT

4.1. Experiment 1

In order to evaluate the accuracy of the mouse movement detection. The first experiment contains three main parts which test various speeds of mouse movement: slow speed, normal speed, and

fast speed. Each test will take different combinations such as slow speed with exceeded boundaries or fast speed with exceeded boundaries. By taking a close look at Table 1, the results supported that the system was sensitive to any moving speed of the mouse that once the mouse has moved outside of any two boundaries, the system will easily detect that the user has moved the mouse out of the testing window. However, the experiment also pointed out an issue of the algorithm that when a mouse was slowly moved out of one edge, the system did not detect this behavior. Recall back to the mouse detection algorithm, the cheating behavior will only detect when the user has touched at least two warnings. Since the user slowly moves the mouse out to a specific boundary, the rapid movement detection did not activate, the only warning was the mouse exceeded one of the boundaries, which caused the system not to be satisfied with the base case resulting in a not detected result. Based on this issue during the experiment, it also brought up a great idea for the improvement of the system, in which the improved algorithm could limit the moving area of the mouse clicker. For example, once the exam has started, the mouse can only move within the testing window, once the mouse clicker touches any boundaries, it will be relocated to the center of the testing window. By doing this, the system easily handles the issue of the boundaries that not only locked the mouse within a certain area but also limit students to only interact with the testing information without checking anything outside of the testing screen.

Table 1. Mouse Movement Detection Experiment Result

Mouse Moving Speed	Left Boundary	Right Boundary	Up Boundary	Down Boundary	Result
Slow	Exceeded	Not Exceeded	Exceeded	Not Exceeded	Detected
Slow	Not Exceeded	Exceeded	Not Exceeded	Not Exceeded	Not Detected
Slow	Not Exceeded	Exceeded	Exceeded	Not Exceeded	Detected
Slow	Not Exceeded	Not Exceeded	Not Exceeded	Not Exceeded	Not Detected
Normal	Not Exceeded	Exceeded	Exceeded	Not Exceeded	Detected
Normal	Exceeded	Not Exceeded	Not Exceeded	Exceeded	Detected
Normal	Not Exceeded	Not Exceeded	Not Exceeded	Not Exceeded	Not Detected
Normal	Not Exceeded	Exceeded	Exceeded	Not Exceeded	Detected
Fast	Exceeded	Not Exceeded	Not Exceeded	Not Exceeded	Detected
Fast	Exceeded	Not Exceeded	Exceeded	Not Exceeded	Detected
Fast	Not Exceeded	Not Exceeded	Not Exceeded	Not Exceeded	Detected
Fast	Not Exceeded	Not Exceeded	Not Exceeded	Exceeded	Detected

By considering different cheating methods and applications, it is necessary to consider various anti-cheating strategies. The system contains a sound detection system in order to prevent students asking help from someone else such as calling a friend or discussing with a tutor. The second experiment created a realistic testing environment to test how the sound detection system works. Three separate empty rooms were prepared for three different students. One laptop was placed on the desk, and students were allowed to take their phones. All three students were required to take the same test. There was no anti-cheating application used for student 1. Student 2 was required to use a lockdown browser system, and student 3 was using our system. As shown in Table 2, it recorded students' behaviors, test results, and cheating detection results during the test. Student 1 had no problem passing the test because no detectors were placed on the laptop. Student 2 faced a little problem searching for resources online, but student 2 decided to ask help from his friend by calling a voice chat using the phone, but still passed the test at the end. Student 3 did the same thing as student 2, but he had no chance to cheat on the test because the sound detection system had received a high amplitude of the sound wave and ended the test immediately. Although student 3 did not use the browser searching for the help, the test would still be ended based on the mouse movement detection. By doing this experiment, the cheating detection system had been working in a good way to prevent cheating behaviors. The results from experiments have reached my first round of development and testing expectations where the detection algorithms could deal with the common cheating methods while students are taking the tests.

Table 2. Sound Detection Experiment Result

Candidates	**Laptop Status**	**Phone Status**	**Test Condition**	**Cheating Detection**
Student 1	Easily searched all the answers on websites	Not Used	Passed	Not Detected
Student 2	Not able to open the browser	Used to call his friend	Passed	Not Detected
Student 3	Taking the test	Used to call his friend	Not Passed	Detected

5. RELATED WORK

Razan B. et al mentioned a computer-vision algorithm to prevent cheating behaviors in online exams [13]. The system takes action to detect the eyes movements of the users by taking images.

This idea is similar to our current face capture system that is still being designed. The system will capture any sensitive movement from the user's face but the algorithms have to be implemented carefully where different factors need to be considered into the algorithms such as eyes focusing time on a specific position or the user moving out of the testing position.

Mohammad M. et al identified a great idea on recording pieces of video during the online exam [14]. Once the students have any prohibited behavior, the system will record that piece of video for future checking. By doing this, it saves a big amount of time from the instructor who only checks small pieces of videos instead of watching the whole recording of how students are doing in tests. Related back to our sound detection system, it focuses on the amplitude of sound waves. Based on the recording idea, it could be helped for the improvement of the sound detection system that the system would record a piece of time for the unusual sounds so that the instructor could double check the detection result based on those short recordings.

Ali M. et al discussed an approach of unsupervised learning that iterates and learns from group data in order to predict cheating behavior while taking the tests [15]. Machine learning plays an important role in handling and predicting results. It would be good to keep implementing machine learning into our mouse movement detection system where the algorithm will be more advanced to catch any students who have the same or similar mouse movements as other detected students. By using machine learning, it not only improves the flexibility of the algorithm but also increases the accuracy of each detection in the future.

6. CONCLUSIONS

Not only has the pandemic of Covid-19 caused many of the classes to switch to the online format, there are still other programs or online courses that have been set as remote in order to provide a more convenient learning experience for other countries' students. However, one of the biggest issues during online education is how to effectively prevent the issue of online academic dishonesty. By making this exam cheating detection system, it helps instructors alleviate the issues of online exams. The system contains various aspects to detect any cheating behaviors happening during the exams. Mouse movement detection restricts students from looking up any prohibited resources through the browser or any other information outside of the testing window. Sound detection supports that the exam is taken by students themselves without having a conversation or discussion with someone else. As the face capture system will be finished later, the system will be more comprehensive and checks any sensitive facial movement to avoid students from looking at resources by hand such as textbooks or notes, and it ensures that the student is not moving on the seat during the exam. According to the experiment, it proves that the

system can effectively prevent the issue of online academic dishonesty that catches any students who are having prohibited behaviors while taking the exam. However, the system does have a deviation in some cases. For example, if the student accidentally touched the mouse causing the mouse to move outside of the boundary with a rapid movement, the system would lock the exam right after it, but the student did not mean to cheat. Although this is a small probability, it is still important to consider how to avoid this kind of mistake. The current idea is that the system needs to be improved to count how long the mouse clicker has been staying outside of the boundaries. By taking this improvement, if the student has accidentally touched the mouse, they could have a few seconds to bring the mouse clicker back to the testing window. The cheating detection will be activated only if the mouse has stayed outside of the testing window for certain seconds. Another solution has been discussed in the earlier section that limits the mouse within a moving area. Once the mouse touches any boundaries, the mouse clicker will be pulled back to the center of the window. In addition, an external feature is also considered to be implemented into the system in the future, where the instructor can decide to set the exam to be the open book format so that the mouse movement detection will be closed because students will be allowed to check any resources. By doing this, the improvement improves the flexibility of the system that provides different formats and restrictions for the exams.

REFERENCES

[1] Benigno, Vincenza, and Guglielmo Trentin. "The evaluation of online courses." Journal of computer assisted learning 16.3 (2000): 259-270.
[2] Ball, Nicholas L., Robert E. Wood, and Gove Allen. "USING TECHNOLOGY TO DETECT AND DETER ELECTRONIC CHEATING AT A LARGE PUBLIC UNIVERSITY." Issues in Information Systems 17.4 (2016).
[3] Pokhrel, Sumitra, and Roshan Chhetri. "A literature review on impact of COVID-19 pandemic on teaching and learning." Higher Education for the Future 8.1 (2021): 133-141.
[4] Ali, Wahab. "Online and remote learning in higher education institutes: A necessity in light of COVID-19 pandemic." Higher education studies 10.3 (2020): 16-25.
[5] Raines, Deborah A., et al. "Cheating in online courses: The student definition." Journal of Effective Teaching 11.1 (2011): 80-89.
[6] Mubin, Omar, et al. "Can a robot invigilator prevent cheating?." AI & SOCIETY 35.4 (2020): 981-989.
[7] Kieslich, Pascal J., et al. "Mouse-tracking: A practical guide to implementation and analysis 1." A handbook of process tracing methods. Routledge, 2019. 111-130.
[8] De Smedt, Tom, and Walter Daelemans. "Pattern for python." The Journal of Machine Learning Research 13.1 (2012): 2063-2067.
[9] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386.
[10] Bawarith, Razan Hamza. "Student Cheating Detection System in E-exams." (2017).
[11] Dufaux, Alain, et al. "Automatic sound detection and recognition for noisy environment." 2000 10th European Signal Processing Conference. IEEE, 2000.
[12] Zarkasi, Ahmad, et al. "Face movement detection using template matching." 2018 International Conference on Electrical Engineering and Computer Science (ICECOS). IEEE, 2018.
[13] Bawarith, Razan, et al. "E-exam cheating detection system." International Journal of Advanced Computer Science and Applications 8.4 (2017).
[14] Masud, Mohammad M., et al. "Smart Online Exam Proctoring Assist for Cheating Detection." International Conference on Advanced Data Mining and Applications. Springer, Cham, 2022.
[15] Duhaim, Ali M., Safaa O. Al-Mamory, and Mohammed Salih Mahdi. "Cheating detection in online exams during COVID-19 pandemic using data mining techniques." (2021).