A MOBILE PLATFORM FOR TEACHERS AND PARENTS TO TRACK CHILDREN'S BEHAVIOR DURING ONLINE CLASSES USING ARTIFICIAL INTELLIGENCE

Ziqi Liu[1], Yujia Zhang[2]

[1] The Governor's Academy, 1 Elm St, Byfield, MA 01922
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Online schooling has become more and more popular during recent years due to COVID-19 [1][15]. It allows teaching to continue without in-person contacts. A prominent issue with online schooling is teachers are unable to oversee students' behavior during class as they would in-person. It has been known that many students tend to lose attention. This can make online schooling less effective, causing it to yield worse results than in-person schooling [2]. In order to tackle this issue,this paper outlines a tool that has been developed to monitor children's mouse and keyboard movements during online classes and analyze the data with artificial intelligence to ensure students are focused in class [3]. For example, if the students are typing and clicking their mouse frequently, then there is a higher possibility the student is not focused because frequent keyboard and mouse movements might indicate they are chatting with friends or playing games; on the other hand, if they are attentive in class, there would be less keyboard and mouse movements, as they should be taking notes.

KEYWORDS

AI, Online learning, Mobile APP

1. INTRODUCTION

Online education has become a second method of teaching as it grew in popularity during the pandemic. Remote learning has also been beneficial for international students, and for classes to reach a wider range of people [4]. During the pandemic, school has been switched to remote and students have less of an incentive to pay attention in class, as there is not a teacher to constantly check on their behavior. This can cause students to learn less and get bad grades. This monitoring tool will send notifications to teachers and parents if their students are not focused, which can assist the teachers' and students can also learn more effectively. The monitor tool has three ends: the student software, an analysis server, and the teacher and parent mobile app. The student software will track the user's mouse and keyboard movements, and send them to our realtime database form Firebase [5]. The analysis server will analyze the data with AI, and then send the results to the parents and teachers. The AI's job is to determine whether the student is attentive or not. The mobile app will receive the results from the AI and send a notification to the teachers and parents if the students are not focused [6].

David C. Wyld et al. (Eds): ICDIPV, CBIoT, ICAIT, WIMO, NC, CRYPIS, ITCSE, NLCA, CAIML -2023
pp. 435-442, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131334

There are not many mouse and keyboard tracking apps to monitor students yet, but many teachers have tried other methods to help make sure students are paying attention. Some teachers try to use some methods to make sure the students are paying attention. One common method is to have students open their cameras. In many online classes, we often hear teachers say, “Please turn on your cameras.” While this method can help the teachers see if the student is present at the class, it is hard to tell the teachers the students are not doing something else on their screen. A tool teachers’ often utilize is to have students share their screens of what they are working on to the teachers so that they can check that the students are actually working. This method is more effective, but the students can always revert back to doing something else after they have stopped sharing their screen. To ensure students are paying attention, teachers also try to interact with students frequently in order to make sure they are focused on the class. For example, the teacher can ask a lot of questions to students so that the students have to pay attention to the class in order to answer them. The problem with this is that it gives the teacher more work as they have to continuously check if the student is listening in class, and it is difficult for the teacher to take care of all the students in the class.

In this paper, we follow the same goal of trying to find a way to make sure students are being attentive in class. Our goal is to eliminate the issues with current methods and tools that were created and used to help students be more productive. The tool that we have will track the student’s keyboard and mouse activity. Their activity will continuously upload the student’s mouse and keyboard movements to the database. The system will provide a notification of its analysis of the student’s behavior immediately to the teacher and parents as soon as the AI detects that the student is frequently typing or clicking, which can be a sign that the student is not focused. This minimizes the issue of teachers having to constantly check on a student to ensure their students are paying attention, as this device will instead send a notification to the teacher if a student seems to not pay attention. This allows teachers to focus on the lesson plan they had planned. The issue of unable to see what the students are doing behind their screen is also avoided because the tool doesn’t necessarily need their screen to determine; all it needs is just the students’ movement. If the student is performing activities on their device that is not needed, their movements will be analyzed and will show that the student is not focused in class.

The structure of the paper will follow 6 different sections. Section 2 will discuss how we started the project and a brief summary of what we did to create the project. Section 3 will talk about the challenges we met and how we solved them. Section 4 will talk about the evaluations that were performed to check if the different components of the system are working. Section 5 will address related works to keyboard and mouse activity tracking. In the last section, the concluding statements point out the future of the system.

2. CHALLENGES

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

2.1. Writing the Tracker Program

Writing the tracker codes was a challenge as there was a lot of time that was allocated into it. First obstacle was to learn about the different packages that could have been used to track activity through a mouse and keyboard. We had to read the documentation of pynput, which is a library that allows us to track mouse and keyboards, and the documentation of tkinter which allows us to create a small GUI (Graphical User Interface) [7][8]. Other than these two libraries, I also had to learn about Firebase. I have never interacted with these two libraries and Firebase prior to this project, so it was difficult for me to learn how to use them and create a big tracking program.

2.2. Linking Parent/Teacher Accounts with Student Accounts

Linking user accounts is another challenge. The system requires three different users, we created accounts for each user; teachers, parents, and students. We needed a way to relate a user to one another in order to send the data and analysis to the appropriate account. We first thought that we could let parents or teachers create an account for their children and connect them by putting the children's user ID under the parents or teachers. But due to the restrictions in Thunkable, which is the website we are using to create our mobile app, in Firebase, and JSON, we gave up this idea. We then tried to use the authentication system in Firebase so that all our users' accounts will be stored in the Firebase authentication system; however, Thunkable didn't like the idea and the app crashed everytime we tried to create a student account and connect it to a parent account.

3. SOLUTION

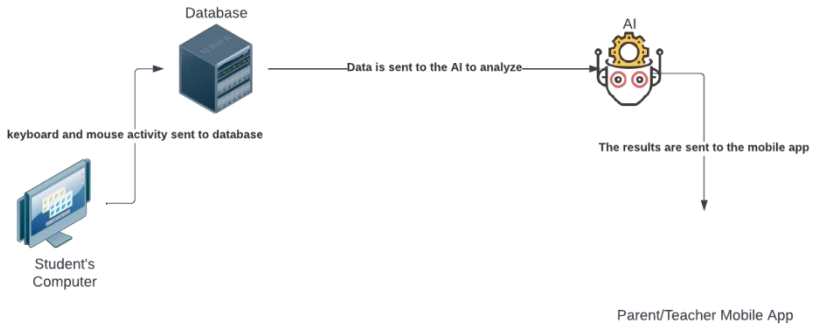

Figure 1. The Structure of the System

Figure 1 represents the overall system of the tool. The first part of the system is to track the student's activity on the computer, which would be the student's computer in Figure 1. We track the student's behavior by using the tracking program we developed. When the student logs in, the program will start recording the position of the mouse when it moves, scrolls, and clicks. It will also record the keys that are pressed. The computer will send these data to our realtime database. The tracking program is made with Python, and we are using pynput as our library to track the MnK (Mouse and Keyboard) activity. We are also using other libraries like Tkinter to create a GUI for the program, and also the requests library to send data into our database.

The second part of the program is the AI analysis program. This program will read the MnK data in the database, and then determine the student's level of activity. For example, if the student is focusing on the class, then the MnK movement should not be too much. However, if the student pressed a lot of keys, then it may indicate the student is chatting with someone. Once an analysis is made by the AI, it will send its final verdict to the parent and teacher mobile app. The parent or teacher will get a notification if the AI believes that the student is not being attentive.

The first component is the tracking program, which is made with Python. We used libraries such as pynput, requests, Tkinter, and Firebase to implement this component. Pynput is a library that allows the system to listen into the user's keyboard and mouse activities. This library was primarily used to grab user's keyboard and mouse information. We first created a Tracker class so we can make a tracker object. In the Tracker class, there are four major functions.

```
def on_move(self, x, y):...

def on_click(self, x, y, button, pressed):...

def on_scroll(self, x, y, dx, dy):...

def on_press(self, key):...
```

Figure 2. Tracker Functions

In Figure 2, the first function will track the movement of a user's mouse (def on_move()), second function listens to every click a user's mouse does (def on_click()), third function tracks if a user is using the scroll wheel of the mouse (def on_scroll()), and the last function is to record if a user is pressing on their keyboard (def on_press()). In order to keep track of the user's activity we used Pynput. Tkinter is Python's library to create a simple GUI (Graphical User Interface) [9]. Figure 3 demonstrates that we created a basic window that the user can use to start the tracking program.

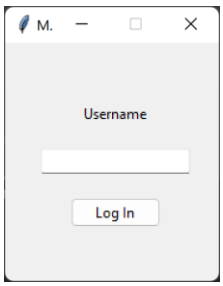

Figure 3. User's Interface

```
def check_connection():...

def verify_user():...

def end_tracking():...

def on_closing():...
```

Figure 4. User Interface functions

The user would either need to create an account or log into an account that was previously made in order to start the tracking program, shown in Figure 3. Figure 4 contains the functions that we used to create the window. We first check for internet connection which is handled using the check_connection()function, because we require the internet to upload the data onto our database. Then, we verify the user with their user ID using the verify_user()we created. If the user is in our system, then the tracking program will start. The end_tracking() function will end the whole tracking program when the user clicks on end, and to add a safety feature we also created the on_closing() function so that when the user directly clicks on "x" and closes the window, the data will still be uploaded. Once the tracker system has started, the activity of the user's mouse and keyboard will be recorded as long as the window is still open. When the tracking program is reading the user's activity, it will use the requests library to upload the data collected to our

realtime database provided by Firebase. When we are uploading the data, we need to convert our data into JSON so that Firebase can accept them.

```
response = requests.put(f'{url}/tracker/{userID}/{action}.json', data=json.dumps(record))
```

Figure 5. Converting Data to JSON

In Figure 5, there is a line of code, which was used to convert our data into JSON, in order for the database to interpret, then send it into our database [10].

4. EXPERIMENT

4.1. Experiment 1

I tested how the AI responds to mouse activity first. I chose a few numbers that are fairly far away from the threshold which is 25, and then I chose 24 which is one less than the threshold and also 25 which is right on to see if the AI can detect the small differences. I also tried both numbers 3 times to make sure the AI makes no mistake. Then I picked some numbers above the threshold just to make sure everything is working properly.

Mouse clicks	AI Conclusion
0	Focused
5	Focused
24 (3 times)	Focused (all 3 times)
25 (3 times)	Not focused (all 3 times)
30	Not focused
73	Not focused

Figure 6. Result of experiment 1

The result of the experiment shows that the AI is completely capable of determining whether the student is listening based on the number of mouse clicks. It reported accurate and precise results for all test cases including the ones that are supposed to challenge the AI.

4.2. Experiment 2

The second experiment is to test how the AI responds to keyboard activities. The AI should be able to identify if a stream of characters is a dialog, or detect if the student is playing games if a student is pressing 'WASD' frequently. Making conclusions from keyboard data is more complicated, so I chose to use a variety of test cases to challenge the AI. I mixed up some regular test cases with some challenging ones to test if the AI can adjust.

Characters Typed	Result	Reason
w	Focused	N/A
Hi	Not focused	Chatting
adjskcasd	Not focused	N/A
wdasadwaswaddaswwa	Not focused	Playing games
	Focused	N/A
This class is so boring	Not focused	chatting
cc	Focused	N/A
w a s d	Focused	N/A
wasd	Not focused	Playing games

Figure 7. Result of experiment 2

The results of the experiment demonstrate that the AI can give the correct conclusion and reason almost every time, except for test number 8. The AI seemed to be unable to tell that the student is playing games with the spaces between the characters. This is crucial information to me as I now know a bug in the AI that I should fix. Other than 'WASD', the AI can detect that the students are chatting.

Both experiments confirm that the AI is performing as I expected; however the AI is having some issues with 'WASD' keys, and this is what I aim to solve. Other than that, the AI can determine that the student isn't focused if they click their mouse frequently or chat with others. The experiments are very helpful as they show some minor flaws in the AI that I should fix. Overall, the AI is working properly as it can report correct behavior of the students based on their keyboard and mouse activity: it concludes that the student is focused every time the number of mouse click is below 25, and vice versa when the number is above 25; it also reports that the student is chatting when words or phrases are typed, or playing games when 'WASD' keys are pressed.

5. RELATED WORK

The authors of this paper want to track mouse movements in order to diagnose "technology acceptance items for students when interacting with a web-based tutoring system during a web development course" (Tzafilkou & Protogeros, 2020), this means the author is finding a correlation with technology and online tutoring [11]. Despite that their ultimate goal is different from us, we are both trying to track user mouse movements to analyze their behavior. Other than the mouse, we also track keyboard movements to assist us in analyzing student behavior, so that we can get more accurate results. Between the two works there are slight differences in the developing phase. In the paper, they state that they are using JavaScript to create their mouse tracking tool, and they integrated the program into learning platforms. We used Python to create our tracking tool, and we are also using AI to analyze the data for us. So, our tool is an independent application.

The researchers are using biometric and machine learning methods to track students' attention and engagement [12]. While this paper's aim is the same as ours, their approach is a bit different. They are using biometric methods like facial recognition, voice recognition, and eye tracker, while we are only relying on mouse and keyboard. In their research, they noticed " challenges arise in online classrooms, which often limit instructors to watching students' body language in video feeds, where they cannot see, for example, distractions in the students' environment" (Villa et al., n.d.). This led to looking for features to monitor in order to help decrease these challenges. Though this project and the one I proposed are different, perhaps in the future, we may add

biometric features into our tool to make it even better. If we are able to combine the differences of our projects, we can create a more powerful tool that can provide accurate reports on students' attention during online learning. Machine learning is something we both have in common. We are also utilizing AI and machine learning to analyze the data we collect.

The authors of this paper are tracking mouse movements, but on overseas quizzes [13]. Their mouse tracking system was "implemented on the Moodle learning management system and tested on an online quiz session accessed abroad" (Purnama et al., 2020). They were also able to record their data in real-time. Overall, their project is very similar to ours, since we are also tracking mouse movement real-time, but our platform is a bit different. We put more focus on online learning, especially in online classes, but it doesn't mean that we won't explore the area of online quizzes and tests. Perhaps later on, we can update our tracking tool so that it can monitor students during quizzes and tests, because it is relatively easier to cheat online. Research like this can always bring us new ideas on how to make our program better.

6. CONCLUSIONS

In this paper, we propose a monitoring tool for teachers and parents to help their children focus during online learning. With this tool, teachers and parents no longer have to worry about their child during online lessons because this tool will do the monitoring job for them. We will send a notification to teachers and parents whenever the AI suspects that the student is doing something other than listening to the lesson.

The current limitation is that we can only track keyboard and mouse activity, and we can only rely on these two factors to determine the student's status. There may be some misjudgement on the keyboard activity because sometimes chatting and taking notes can have very similar keyboard movements. This is the same for mouse movements, because some students will have to move their mouse when performing some tasks, or reading notes.

Another optimization problem is that this program may cause lags on some low-specification computer, or in places with bad internet. This tool requires a strong internet connection because it is constantly uploading data to a cloud database. During the upload, it may cause some lag on some computers.

We can add some new features to the app. For example, we can look at the user's camera, and record their screen to get more accurate results. But, one problem with this is that it may violate people's privacy and some people don't want this to happen. If we look at the user's camera, we will be able to analyze the user's eye contact with the screen and their overall facial expressions [14]. If we are able to get the user's screen, it will be even easier for us to tell the parents and teachers whether their student is focused or not, but not necessarily everyone would like their screen to be recorded as this may violate people's privacy.

One way to optimize the application so it will cause less lag is to let the tracking program upload the data less frequently, which means a bigger time gap between each upload. The first time we tested the tracking program, which would upload data every time we move the mouse or touch the keyboard, the program became laggy. So, we think putting more time between each upload may reduce the lag, but at the same time it may cause less on-time notification.

REFERENCES

[1] Beck, Dennis, and Jason LaFrance. "Online schooling in the United States: A response to Saultz and Fusarelli." Journal of School Choice 11.1 (2017): 42-59.

[2] Dykman, Charlene A., and Charles K. Davis. "Part One: The Shift Toward Online Education." Journal of Information Systems Education 19.1 (2008).

[3] Hamet, Pavel, and Johanne Tremblay. "Artificial intelligence in medicine." Metabolism 69 (2017): S36-S40.

[4] Hobbs, Tawnell D., and Lee Hawkins. "The results are in for remote learning: It didn't work." The Wall Street Journal (2020).

[5] Moroney, Laurence. "The firebase realtime database." The Definitive Guide to Firebase. Apress, Berkeley, CA, 2017. 51-71.

[6] Joorabchi, Mona Erfani, Ali Mesbah, and Philippe Kruchten. "Real challenges in mobile app development." 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE, 2013.

[7] Lundh, Fredrik. "An introduction to tkinter." URL: www. pythonware. com/library/tkinter/introduction/index. htm (1999).

[8] Jansen, Bernard J. "The graphical user interface." ACM SIGCHI Bulletin 30.2 (1998): 22-26.

[9] Beniz, Douglas, and Alexey Espindola. "Using Tkinter of python to create graphical user interface (GUI) for scripts in LNLS." WEPOPRPO25 9 (2016): 25-28.

[10] Pezoa, Felipe, et al. "Foundations of JSON schema." Proceedings of the 25th International Conference on World Wide Web. 2016.

[11] Purnama, F., Sukhbaatar, O., Choimaa, L., & Usagawa, T. (2020). Implementation of real-time online mouse tracking on overseas quiz session. Education and Information Technologies, 25(5), 3845-3880. https://doi.org/10.1007/s10639-020-10141-3

[12] Tzafilkou, K., & Protogeros, N. (2020). Monitoring mouse behavior in e-learning activities to diagnose students' acceptance items of perceived usefulness and ease of use. The European Educational Researcher, 3(1), 21-27. https://doi.org/10.31757/euer.312

[13] Villa, M., Gofman, M., Mitra, S., Almadan, A., Krishnan, A., & Rattani, A. (n.d.). A survey of biometric and machine learning methods for tracking students' attention and engagement. 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 948-955. https://doi.org/10.1109/ICMLA51294.2020.00154

[14] Senju, Atsushi, and Mark H. Johnson. "The eye contact effect: mechanisms and development." Trends in cognitive sciences 13.3 (2009): 127-134.

[15] Ciotti, Marco, et al. "The COVID-19 pandemic." Critical reviews in clinical laboratory sciences 57.6 (2020): 365-388.