BrainBridge: A Browser Extension that Improves Online Learning with Brain Data

A Major Qualifying Project (MQP) Report
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements
for the Degree of Bachelor of Science in

Computer Science

By:

David Danielian

Project Advisors:

Erin Solovey
Alicia Howell-Munson

Date: May 2023

Abstract

Brain-Computer Interfaces (BCIs) show promise in helping enhance the way we use software. By using machine learning models, it is possible to identify states such as proactive control, reactive control, rule acquisition, rule-following, mind-wandering. In the field of education, having insight into the cognitive state of the user means we can better understand why students are struggling. In this project, we seek to bridge the gap between online learning platforms and brain data, to enable the creation of BCI that utilize brain data to improve students learning outcomes. After conducting a persona study and discussing with researchers, we created BrainBridge a browser extension and API which facilitates communication between online learning platforms and brain data neuroimaging headsets. BrainBridge is capable of visualizing and providing real-time interventions based on a users cognitive state, while also logging actions in the learning platform and mapping them to brain data. BrainBridge was built and tested with the ASSISTments learning platform and fNIRS neuroimaging cap in mind, but has the potential to be expanded to support other neuroimaging techniques and online learning platforms.

Acknowledgements

- We'd like to thank Shreya Mhalgi, Deniz Sonmez Unal, Theresa G. Mowad, Erin Walker, and Kate Arrington for their valuable feedback on the researcher tools
- This work was supported in part by the U.S. National Science Foundation under Grant NCS-1835307

Contents

List of Tables

List of Figures

1 Introduction

Brain-Computer Interfaces (BCIs) are systems that react to brain data from the user. BCI has been the subject of a variety of research in fields such as medicine [1], marketing [2], and gaming [3]. One field that stands to benefit from BCIs is education. Teachers may not have the time or resources to make sure that all of their students fully understand the material. Traditional methods of gauging understanding, such as exams or homework, are flawed. Exams are prone to causing anxiety [4] and studies show that for a significant portion of students they are not an effective measure of understanding [5]. These problems are only amplified with digital learning platforms where it is less feasible to provide a personalized learning experience.

Monitoring the user's brain activity can provide educators with direct input on the cognitive state of students, giving insight on what might be causing them to struggle. BCIs have been used previously to assess academic performance by measuring the attention of students during learning tasks [6]. While assessing performance is helpful, there is a need for BCI that can provide real-time feedback and meaningful interventions to students during learning.

Researchers at Worcester Polytechnic Institute's Human-Computer Interaction Lab (WPI HCI Lab) have been working towards creating machine learning models that classify various cognitive states from brain data collected from Functional Near-Infrared Spectroscopy (fNIRS) devices. This research is specifically aimed towards classifying states involved with learning tasks. These include rule acquisition and rule following [7]. With progression towards identifying these states in realistic learning environments, we need to assess how students and educators can leverage this information to improve learning outcomes.

The goal of this project was to create a BCI that can monitor the user's brain data in real time, classify it, and visualize their cognitive state. We created a browser extension which tracks the user's activity, maps it to their brain data, displays their cognitive state, and creates interventions as they perform learning tasks. In addition, this proof-of-concept system serves as a platform for rapidly prototyping interventions and refining machine learning models to advance the educational BCIs.

2 Background

2.1 Functional Near-Infrared Spectrocopy

Functional Near-Infrared Spectroscopy (fNIRS) is a neuroimaging technique that measures brain activity by tracking blood oxygenation levels in the user's head. fNIRS has become a popular method of non-invasive neuroimaging among researchers [8]. Source optodes arranged on the fNIRS cap emit near-infrared (NIR) light into the scalp which is then picked up by detector optodes [9]. The diffusion of the NIR light received by the detectors correlates with changes in the concentration of oxygenated hemoglobin, which over time indicates neural activity [10].

In studies that require wearable, non-invasive neuroimaging techniques, the most accessible choices to researchers are electroencephalography (EEG) and fNIRS. A variety of factors make fNIRS an effective option for educational settings. Relative to EEG, fNIRS has a higher spatial resolution, allowing it to measure localized brain activity [11]. This makes fNIRS more effective at gauging region-specific cognitive states.

fNIRS caps are small, lightweight, and comfortable. Setting up the cap is relatively simple. No gel or special equipment is required to get a good signal. fNIRS has also been shown to be resilient to movement during data collection due to noise filtering techniques such as artifact detection and band-pass filtering [9, 11]. This makes it a good choice for studies that may involve movement, such as using multiple displays or performing mouse and keyboard operations. One limitation of fNIRS is the inherit delay in the measured hemodynamic response. It typically takes around 5 seconds after a stimulus to register the brain activity while EEG is near instantaneous [11]. However, a delay of a few seconds is insignificant in an educational environment where study sessions can range from several minutes to hours. The ability to accommodate a wide range of users comfortably makes fNIRS a promising choice for studies on education.

2.2 Brain-Computer Interfaces

Brain-Computer Interfaces (BCIs) are systems that use brain-data from a user to interact with a digital environment. This involves a device to measure brain activity and an interface that creates a meaningful response to the brain activity. BCIs are often classified as active or passive [12]. Active BCIs detect intentional input from the user in order to trigger an output; an example of an active BCI is the use of brain data to move a computer cursor to a desired position [13]. These BCIs are typically used as medical augmentations to assist those with impaired motor function, as this allows the system to detect user intention without physical input. One application researchers work towards is using BCI to control prosthetic limbs which can restore a user's motor capabilities [1]. Passive BCIs, on the other hand, listen for implicit brain activity and react accordingly. Passive BCIs are often used to monitor a user's cognitive state and intervene when certain states are reached. One study created a passive BCI that effectively detected drowsiness during driving simulations [14]. This type of BCI is of particular interest in the field of education, since it has potential to provide valuable feedback on the cognitive state of students while they learn.

2.2.1 BCI in Education

Many researchers have been studying how BCI can measure and potentially improve learning experiences. One study was able to use an EEG-based BCI to measure and train away math anxiety in students, increasing their performance [15]. Another study successfully correlated the effectiveness of learning with attention levels tracked by an EEG headset [6]. While measures of emotion and attention give insight on whether the student is focused on the task, they do not necessarily give insight on the inductive reasoning process of the student [16]. Measuring states such as rule acquisition, rule following, and proactive or reactive learning gives more direct insight into how a student is learning and whether they will retain information [7, 17]. These cognitive states would give educators more meaningful data to work with when examining learning outcomes.

2.3 Human-Computer Interaction Design Principles

Human-computer interaction (HCI) design plays a crucial role in introducing novel technologies, such as BCI, to users. A goal of the BCI we were designing was that itmust be easy to understand for the user to have a positive experience with the system. One widely accepted set of guidelines for designing interfaces are Nielsen's 10 heuristics, listed in table 1 [18]. Rather than a strict guideline, these heuristics represent general rules of thumb when designing an interface with users in mind. They are commonly used in HCI to evaluate the effectiveness of interfaces.

The incorporation of a user's brain data poses many unique issues that require careful consideration when designing BCI. Due to the noisy nature of brain data collection, data gathered and conclusions drawn from this data can be error prone. This problem is only amplified with machine learning algorithms which operate on probabilities to draw conclusions [19]. Visualizations provided by the interface can also have unintended effects on the users cognitive state. In active BCI, this is known to lead to to what is often

1	Visibility of system status
2	Match between system and the real world
3	User control and freedom
4	Consistency and standards
5	Error prevention
6	Recognition rather than recall
7	Flexibility and efficiency of use
8	Aesthetic and minimalist design
9	Help users recognize, diagnose, and recover from errors
10	Help and documentation

Table 1: Nielsen's 10 heuristics

referred to as the "Midas Touch" problem[20]. We predict that the implicit signals of passive BCI can too be impacted by visualizations. When the user sees a prediction of their cognitive state, it may cause confusion if the system and their perception don't align. This may distract from the task at hand, negatively affecting the user's experience. External stimuli such as the user's attitude and environment can lead to unintended input.

Due to these issues, error prevention and recovery is critical in designing BCI. One method to rule out error is to incorporate data outside of the brain, such as application state, to provide more context to the brain data [20]. It is also important that any interventions from the BCI are either unperceived by users, or clearly communicated and reversible to prevent frustration. To prevent the Midas Touch problem, visualizations need to to be minimal and easy to understand. Research shows that different visualizations affect users differently [21] so flexibility in visualizations could also improve a user's experience.

2.4 Intelligent Tutoring systems

Intelligent tutoring systems (ITSs) are digital platforms that provide coursework, solutions and hints without intervention from the teacher. The effectiveness of ITSs is well documented; they have been shown to be nearly as effective as human tutors [22]. Without guidance from a teacher, ITSs often rely on student control to provide feedback and hints. However, students often lack the meta-cognitive skills to understand when they need help and what kind of help they need [23]. Another challenge with ITSs is understanding context and social queues such as body language. These problems could be helped if the ITS had more data to create meaningful interventions. Previous ITSs have read facial expressions to try to gauge the emotional state of the user and react accordingly [24]. In this research, we believe that brain data can be integrated with ITS to make up for these limitations. This way ITSs can provide meaningful interventions to the user's cognitive state.

2.4.1 ASSISTments

ASSISTments is an ITS built to enhance online education for researchers, teachers, and students alike. It was designed to help students with problem sets while providing teachers and researchers feedback that will help them improve the learning experience. Teachers can track student activity to catch potential misunderstandings and gaps in knowledge [25]. ASSISTments is also often studied and used in research for controlled experiments involving ITS and online learning [26, 27, 28]. On the student side, ASSISTments acts as an ITS with capabilities to assess correctness, provide hints, or break problems into steps when a student is struggling. ASSISTments' researcher friendliness and tools for creating problem sets with meaningful interventions made it an ideal choice for our experiments.

2.5 Educational Browser Extensions

Browser Extensions allow users to expand the functionality of the web browser. They are commonly used to enhance the appearance of their browser, enhance online shopping, check grammar when typing, and much more. As the web becomes an integral part in students' education, extensions have emerged with the goal of assisting online education [29, 30]. Browser extensions have also been used to create accessibility features into browsers such as text-to-speech options, web-page readability tools, and translations [30]. One such extension even integrated an EEG-based BCI to help disabled users interact with their browser [31]. We set out to create a browser extension that will support various applications of BCI research in education to create better learning experiences.

3 Design and Implementation

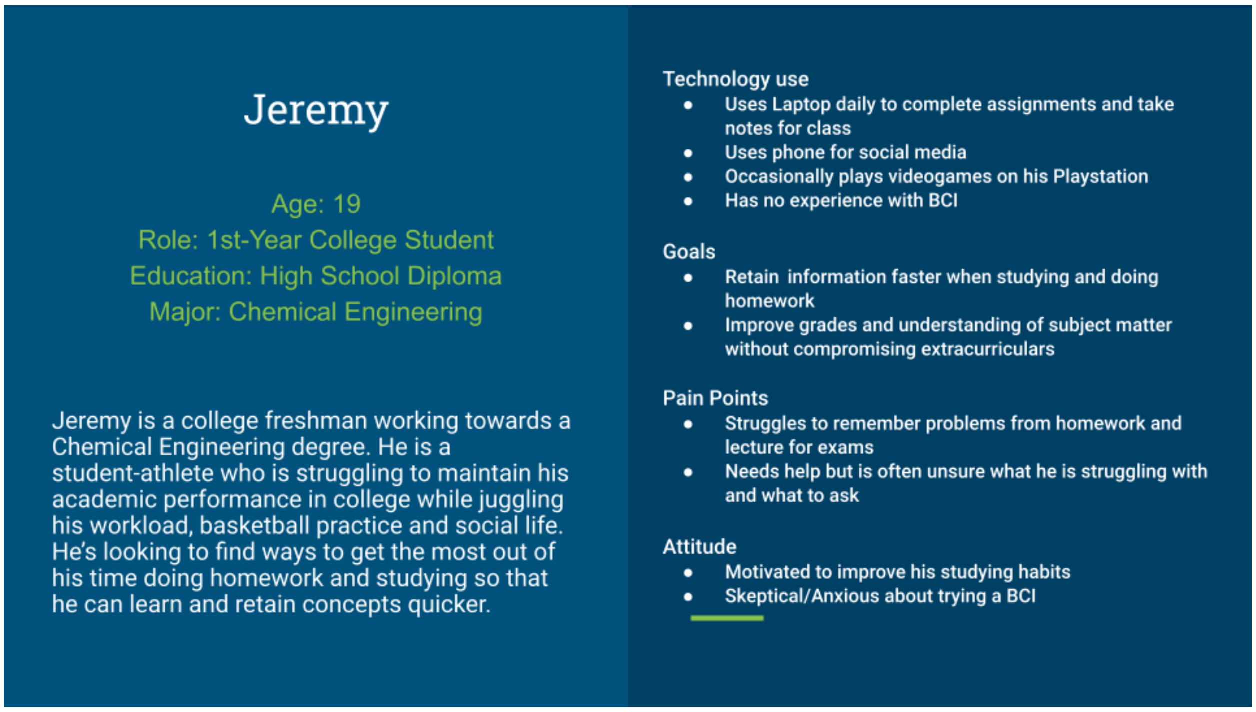
To assess the required functionality of our browser extension we conducted a persona study and talked to researchers working on relevant projects. We chose to create a Google Chrome extension since Chrome is the most widely used browser globally [32]. We created an API using the Flask python library to send data between the browser extension and the brain data. The browser extension interfaces with the ASSISTments ITS, with the goal of assisting and monitoring students completing problem sets on the platform.

3.1 Persona Studies

In human-computer interaction, personas are fictional characters created to represent a population. Introduced to HCI by Alan Cooper, personas are used as a guideline for designing interfaces and encouraging goal-oriented design [33]. By having a single generalized archetype that represents the target audience, designers can better put themselves in the shoes of the end-user and design around their experience. To create an effective interface, it is important to consider the needs of the end user. By conducting persona studies of the target audience, we can better understand what features would be helpful for the end users of the BCI.

To conduct a persona study, we created personas for the populations that can benefit from an educational BCI. We identified three potential users: students, teachers, and researchers. We created one persona for each of these demographics. To do this, we first identified the use case of an educational BCI for each demographic. For example, a student could be interested in using a BCI to foster better studying and homework habits through understanding their cognitive states. Once we established the use case we worked backwards to understand the goals that the BCI will help them attain, as well as the pain points that it will help to resolve. Next, we assessed the experience that we expect the personas have in technology. For example, a younger student who grew up using computers and smartphones will likely be more proficient in navigating a computer than an older teacher. Lastly, we filled in the blanks to get more specific, defining the age, education level, and attitudes of the persona and writing a bio for the persona to get a more intimate understanding of their needs. As development of the BCI happened, we used the personas to discuss why features may or may not be a good idea, such as "Professor Doe wouldn't intuitively think to click here for the menu."

Each persona had distinct needs which require to their own interfaces and features. The teacher persona, Amy, is the least technologically proficient and may find another online tool more frustrating than helpful (Figure 2). Our student persona, Jeremy, wants to improve his studying habits and pinpoint where they might be struggling (Figure 1). To him, it is important that the extension can help him identify patterns in his cognitive state as he completes problem sets. With time being a big concern for Jeremy, interventions to time box him when he is struggling on a problem too long would be a great help. He also is not experienced in analyzing brain data, so we need to visualize the data in an accessible way to him. The researcher persona, Elliot, also tracks a user's cognitive state during learning tasks (Figure 3). The researcher is experienced in conducting BCI studies, which includes using software such as Aurora and Lab Streaming Layer(LSL) for data collection. Aurora in particular has a feature where points of interest in brain data can be marked using


Jeremy

Age: 19
Role: 1st-Year College Student
Education: High School Diploma
Major: Chemical Engineering

Jeremy is a college freshman working towards a Chemical Engineering degree. He is a student-athlete who is struggling to maintain his academic performance in college while juggling his workload, basketball practice and social life. He's looking to find ways to get the most out of his time doing homework and studying so that he can learn and retain concepts quicker.

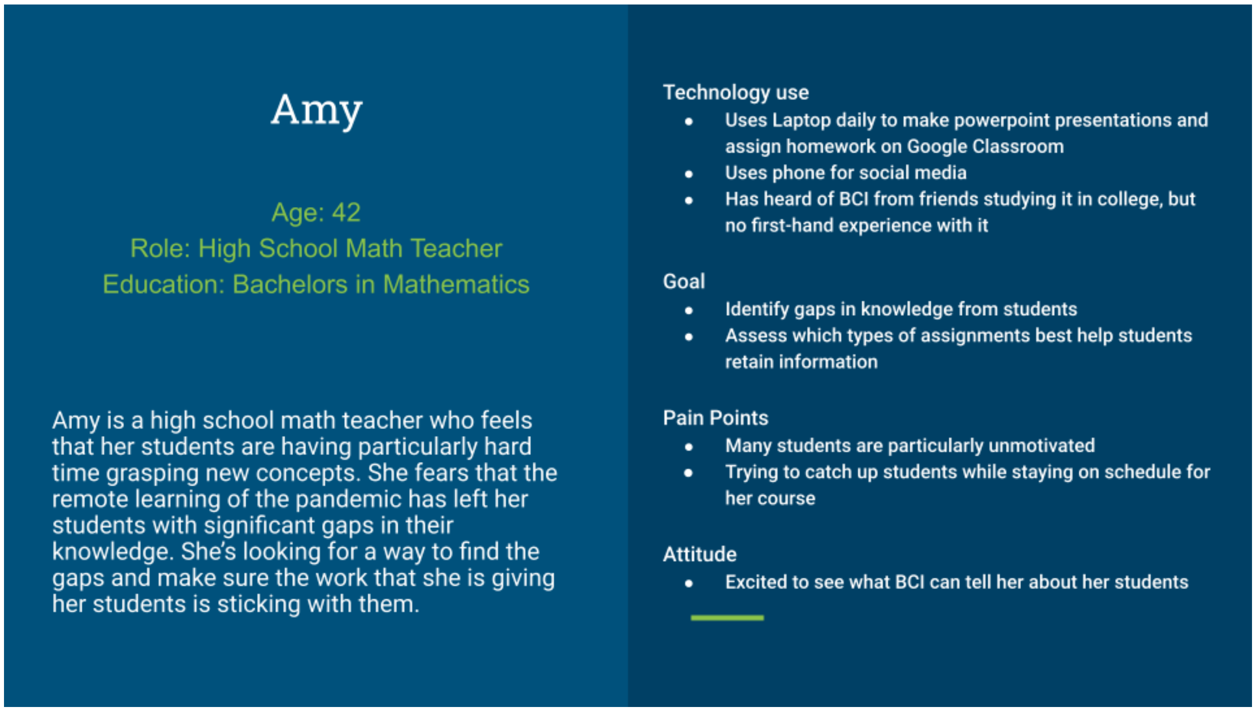
Technology use
- Uses Laptop daily to complete assignments and take notes for class
- Uses phone for social media
- Occasionally plays videogames on his Playstation
- Has no experience with BCI

Goals
- Retain information faster when studying and doing homework
- Improve grades and understanding of subject matter without compromising extracurriculars

Pain Points
- Struggles to remember problems from homework and lecture for exams
- Needs help but is often unsure what he is struggling with and what to ask

Attitude
- Motivated to improve his studying habits
- Skeptical/Anxious about trying a BCI

Figure 1: Student Persona: Jeremy is a student athlete that wants to efficiently use the limited time he has to study.


Amy

Age: 42
Role: High School Math Teacher
Education: Bachelors in Mathematics

Amy is a high school math teacher who feels that her students are having particularly hard time grasping new concepts. She fears that the remote learning of the pandemic has left her students with significant gaps in their knowledge. She's looking for a way to find the gaps and make sure the work that she is giving her students is sticking with them.

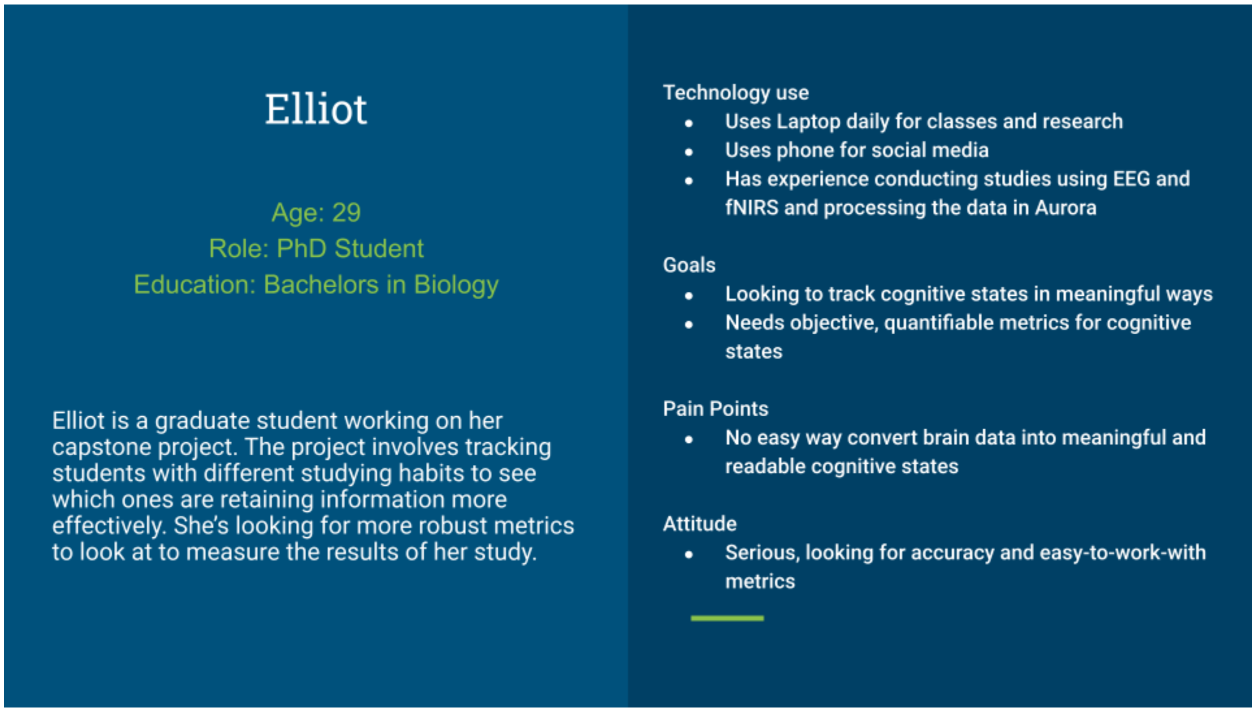
Technology use
- Uses Laptop daily to make powerpoint presentations and assign homework on Google Classroom
- Uses phone for social media
- Has heard of BCI from friends studying it in college, but no first-hand experience with it

Goal
- Identify gaps in knowledge from students
- Assess which types of assignments best help students retain information

Pain Points
- Many students are particularly unmotivated
- Trying to catch up students while staying on schedule for her course

Attitude
- Excited to see what BCI can tell her about her students

Figure 2: Teacher Persona: Amy is a school teacher who needs tools to understand why her students may be struggling and how she can cater her teaching to their needs.



Elliot

Age: 29

Role: PhD Student

Education: Bachelors in Biology

Elliot is a graduate student working on her capstone project. The project involves tracking students with different studying habits to see which ones are retaining information more effectively. She's looking for more robust metrics to look at to measure the results of her study.

Technology use

- Uses Laptop daily for classes and research
- Uses phone for social media
- Has experience conducting studies using EEG and fNIRS and processing the data in Aurora

Goals

- Looking to track cognitive states in meaningful ways
- Needs objective, quantifiable metrics for cognitive states

Pain Points

- No easy way convert brain data into meaningful and readable cognitive states

Attitude

- Serious, looking for accuracy and easy-to-work-with metrics

Figure 3: Researcher Persona: Elliot needs easy to use software to help her get results for her studies on online learning.

triggers. The ability to connect with Aurora and being able to send and easily map triggers to problems in an online problem set would allow her to conduct studies that she couldn't before.

3.2 Desired Functionality

During this project we worked closely with researchers to create features in the browser extension that would help with their research. Researchers at University of Pittsburgh and WPI are working to classify cognitive states during learning tasks [34]. Our browser extension was needed to communicate between the learning software, ASSISTments, and the cognitive state classifications. We asked them what features would be useful to get meaningful results. They expressed the need to capture interactions in the web page that indicate certain cognitive states. For example, the student clicking the hint button likely indicates confusion, logging the time of this would allow the researcher to map this to the brain data. To supplement this, they also proposed a "follow-up" question prompt where users can self report their confidence and thought process while solving a problem.

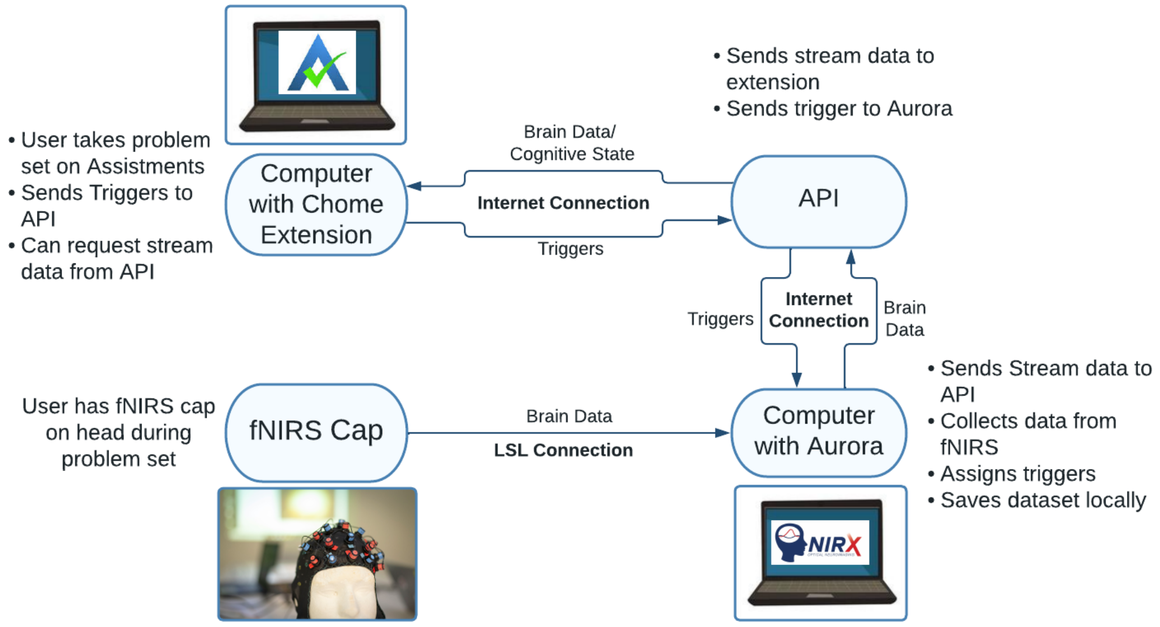
From assessing our personas and discussions with researchers, we concluded that the browser extension requires the following functionality:

1. The ability to display the users cognitive state in real time
2. Easy to understand visualizations of cognitive states
3. Interventions when the user is confused for prolong periods of time
4. Mapping cognitive states to actions on the ASISTments web page as they complete a problem set
5. The ability to send triggers via Lab Streaming Layer (LSL)
6. Question prompt to query the user during a study

The ability to display an easy to understand visualization of cognitive states will help Jeremy better understand how his study habits affect this cognitive state. Interventions would also help him to get help when he isn't making progress at his desired pace. ASSISTments is often used by researchers in studies involving ITS. By mapping actions in ASSISTments to triggers sent via LSL, researchers such as Elliot can better correlate actions with cognitive states. Lastly, a follow-up question prompt will allow the user to provide feedback to researchers during a study.

3.3 Schema

The BCI will require many components to work in harmony. We created a schema to visualize how everything will work together during an experiment. In an experiment, a user will be using a computer to take the problem set with the browser extension activated. Meanwhile, a fNIRS cap placed on their head will be connected to the researcher's computer, which is running Aurora. An API is needed to communicate between these computers. The schema outlines the purpose of each component, the data they pass between each other, and how they are connected (Figure 4).



• User takes problem set on Assistments
• Sends Triggers to API
• Can request stream data from API
Computer with Chome Extension
Brain Data/ Cognitive State
Internet Connection
Triggers
• Sends stream data to extension
• Sends trigger to Aurora
API
Triggers
Internet Connection
Brain Data
• Sends Stream data to API
• Collects data from fNIRS
• Assigns triggers
• Saves dataset locally
User has fNIRS cap on head during problem set
fNIRS Cap
Brain Data
LSL Connection
Computer with Aurora
NIRX

Figure 4: Schema outlining the functionality of the BCI, arrows represent the data passed between the nodes and in bold is what connects them.

3.4 Development

3.4.1 Continuous Integration

Two Git repositories are used to maintain the code base, one repository contains the API and handles connectivity to the fNIRS stream. The browser extension front-end is stored in a separate repository. Separating the API and front-end helps ensure that each can function individually as we integrate new features during development. This was important because the API and browser extension will be running on separate hardware. See Appendix A for more information and links to the Github repositories.

3.4.2 Flask API

To create the necessary functionality we need to define an API responsible for communication between the browser extension and Aurora. We wrote a REST API in Python using Flask in order to interface easily with the existing Python code to fetch the LSL stream. Flask is a lightweight web framework used to create web applications in Python. We chose flask because its ease of use and flexibility made it a great choice for the basic REST API we needed to create[35].

We created POST requests to send triggers from the browser extension. These requests contain a timestamp of when the trigger occurred; this is so we have the option to correct for any network delays between when the trigger happens and when Aurora receives it. Problem specific triggers also contain the problem ID that corresponds to the trigger. GET requests were made for the browser extension to access the brain data. The API schema is documented in Appendix B.

3.4.3 Trigger Mapping and Detection

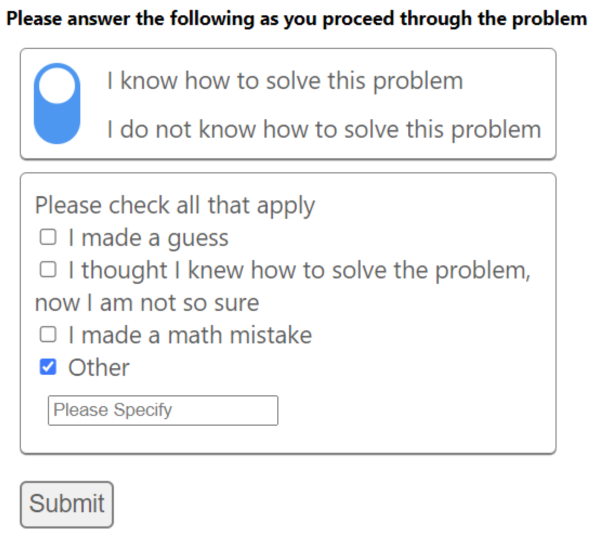
To mark the times of interest during an experiment, we need to send triggers to Aurora when they occur. Aurora accepts triggers as integers, meaning that for each experiment we have to map integers to have a specific meaning. In our case, triggers are mapped to actions in ASSISTments, such as starting a new problem or submitting an answer. Triggers can be made problem specific, for example, submitting problem 1 sends a different trigger than submitting problem 2. To send the right triggers, the mappings need to be assigned in code. This can be done in a separate python file, where dictionaries are defined assigning triggers as a key/value pair corresponding to the ID of the ASSISTments problem. For an example of trigger mapping and how it is defined in the code, see Appendix C.

To detect a user's actions while completing the problem set we can monitor changes on the ASSISTments web page. The Mutation Observer JavaScript API was used to detect new HTML elements on the page. [36] When an element is found containing a new problem, the problem set ID is read and the corresponding start trigger is sent through the API. To detect triggers involving button presses, such as submitting problems or clicking the hint button, we add event listeners to the button's HTML elements to send the appropriate trigger on click.

3.4.4 Logging User Actions

All meaningful interactions with the ASSISTments web page can be logged with timestamps to CSV files. These can be used along with the triggers to map actions with brain data. The feature was implemented using a Node.js server running locally on the computer being used by the student. The researcher is meant to run the server at the start of each experiment, this creates the CSV file with headers for the timestamp and action columns. When the user clicks a button on the ASSIStments page, the local server is called and appends a line to a CSV file logging the timestamp and a description of the action.

Question Prompt User can also self report their confidence in a problem using a question prompt. The question prompt was created to be displayed next to the ASSISTments problem as the user completes it. The prompt contains a toggle switch with two options: "I know how to solve this problem" or "I do not know how to solve this problem." One case of interest among the researchers we talked to was the one where the user was confident that they knew how to solve the problem, but got it wrong. This event is logged and causes a second multiple-choice prompt to appear giving the user a chance to explain what may have gone wrong. All interactions with this prompt are logged similarly to the ASSISTments page. This allows researchers to know the exact time where users become confident in their ability to solve the problem.


Please answer the following as you proceed through the problem

I know how to solve this problem

I do not know how to solve this problem

Please check all that apply

☐ I made a guess

☐ I thought I knew how to solve the problem, now I am not so sure

☐ I made a math mistake

☑ Other

Please Specify

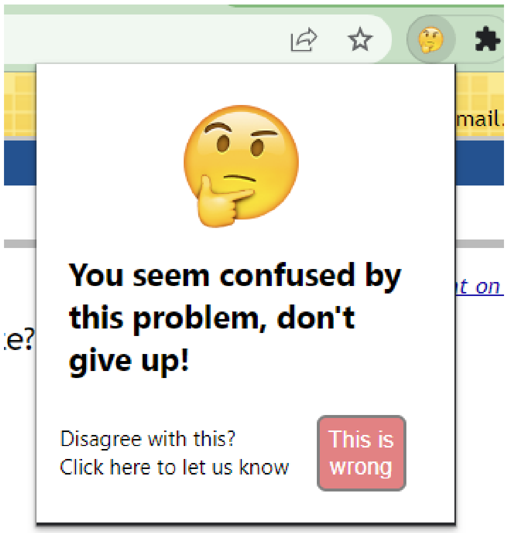
Submit

Figure 5: The question prompt after the user gets the problem wrong with the confidence toggle activated.

3.4.5 Visualizations

Our visualization of the user's cognitive state needed to be immediately recognizable to the user as well as subtle enough to avoid the Midas-touch effect. To achieve this, we used an emoji in the place of the small browser extension icon on the top right corner of the browser. Emojis are commonly used and associated with positive attitudes, especially among young people [37]. This made them an effective choice for providing easy to understand and simple visualizations for students. The browser extension icon is also small enough to be visible, but out of the users way.

The icon updates in real-time based on our prediction of the user's current cognitive state. The user's cognitive state is requested from the API at a sampling rate of 10 times a minute, in order to avoid rapid switching between icons. Users can also click on the icon to open a pop-up menu. In the menu is a brief description of the user's cognitive state as well as a button where they can express if they feel the prediction is incorrect. When the button is pressed, a trigger is sent to Aurora, allowing the researcher to identify instances where cognitive states may have been mislabeled.

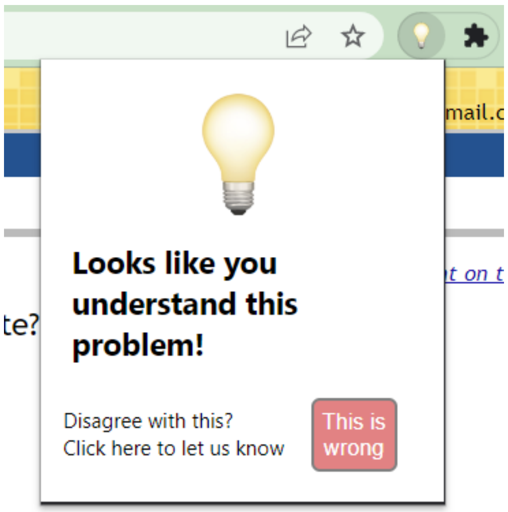


You seem confused by this problem, don't give up!

Disagree with this? Click here to let us know

This is wrong

Figure 6: Expanded Browser extension icon and popup when we predict the user is confused.



Looks like you understand this problem!

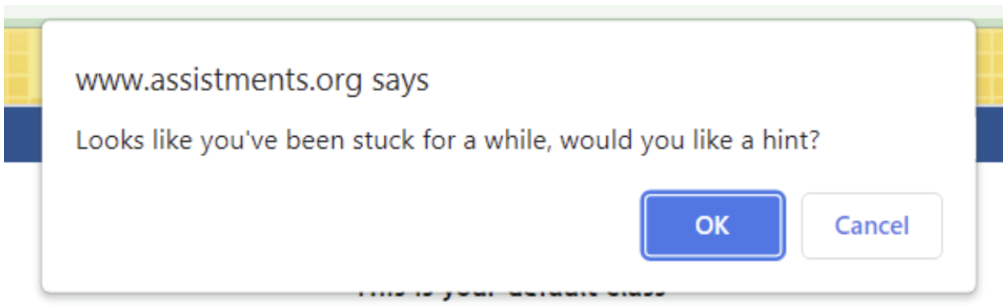
Disagree with this? Click here to let us know

This is wrong

Figure 7: Expanded Browser extension icon and popup when we predict the user understands the problem.

3.4.6 Hint System

To help students who may be stuck on a problem, a hint system was created to intervene when users have been confused for a long time. This would allow our student persona, Jeremy, to time box himself, reminding him to get help if progress isn't being made. To detect prolonged periods of confusion, the user's cognitive states for the last 90 seconds are stored. If the user was confused for 90% of the time, a prompt is shown asking if the user needs a hint. This allows us to know if the user is generally confused while allowing for some error if the cognitive state is misclassified. The parameters of 90 seconds and 90% can be easily changed. Finding the most effective values for these may be a topic of future research. The hint prompt is presented to the user in the form of a alert from the webpage. The user will have the option to accept of decline the hint by pressing the "OK" or "Cancel" button. Upon accepting the hint, the hint button of the ASSISTments problem page is programmatically clicked, as if the user themselves clicked it.


www.assistments.org says

Looks like you've been stuck for a while, would you like a hint?

Figure 8: The hint prompt that pops up in the ASSISTments webpage when the user meets our criteria for confusion.

4 Results

4.1 Implementation

We titled the finished browser extension BrainBridge. It was designed to run on all chromium-based web browsers, however, it was primarily tested and used on Google Chrome. The browser extension was split into two builds, one for researchers interested in data collection and sending triggers, and another focused on real-time feedback for students including visualizations and interventions. The extensions can be loaded into the browser and will automatically communicate with the server once it is running.

4.2 Final API Structure

While the API was originally intended to be hosted remotely, we found that running the API locally was more effective for testing and current use cases. A new schema was created for this work around, see Figure 9. The API can still be hosted on a remote server as long as the API URL is updated accordingly in the browser extension code. The repository for the API is stored in the Neurolearn GitHub repository linked in Appendix A and includes instructions running the API locally.

The API includes four POST requests to send triggers to Aurora from the browser extension. First one is for when a problem is started, second is for when a problem is submitted. Another one is used for when a user steps though a problem after breaking it into steps. The last one indicates that the user disagreed with the current cognitive state prediction. Different triggers can be mapped to specific problems and cognitive states, see Appendix C. A GET request is used to get the rule learning state of the user; this takes data from the neuroimaging cap and processes it through a trained machine learning model [38]. The output is a 0 or a 1 which indicates rule acquisition and rule following respectively. The API schema is specified in

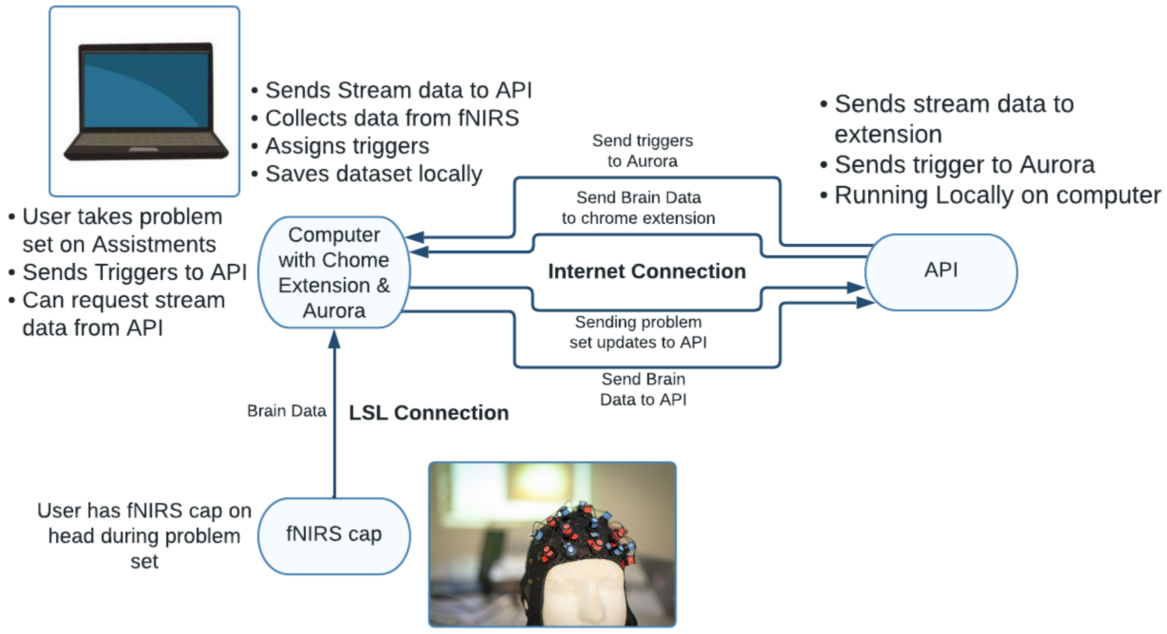

• Sends Stream data to API
• Collects data from fNIRS
• Assigns triggers
• Saves dataset locally
• User takes problem set on Assistments
• Sends Triggers to API
• Can request stream data from API
Computer with Chome Extension & Aurora
Send triggers to Aurora
Send Brain Data to chrome extension
Internet Connection
Sending problem set updates to API
Send Brain Data to API
• Sends stream data to extension
• Sends trigger to Aurora
• Running Locally on computer
API
Brain Data
LSL Connection
User has fNIRS cap on head during problem set
fNIRS cap

Figure 9: Schema for running the API locally on the same computer running ASSISTments and the browser extension. Arrows represent the data passed between the nodes and in bold is what connects them.

greater detail in the API schema: Appendix C

4.3 BrainBridge: Student

This build of BrainBridge provides visualizations and real-time feedback as students work through problem sets. This version contains the real-time visualizations of rule learning states and the automatic hint system. It sends triggers to Aurora as the student traverses through the problem set as well but does not contain other data collection features such as the question prompt and the CSV logging.

4.3.1 Pilot User Studies

To test the interface for bugs as well as get a rough idea of the effectiveness of the classifications and impact of the hint system, we designed an experiment to see what users thought and conducted pilot studies. We recruited 5 college students to be participants in our study (4 male 1 female, M = 22 years old). The participants were tasked with completing an ASSISTments problem set consisting of 9 college level probability problems with the hint system and visualizations. We asked the students about their familiarity with probability to see how this affected the cognitive state predictions. The problem set was designed to be about 15 minutes in length. Each problem could be broken down into steps, with each step having hints and the option to reveal the answer as a last resort. It was explained that the user could use these assistive features at any time, but the hint system would prompt the user if it detected the user spend 90% of the previous 90 seconds of the problem in the rule acquisition state. We also explained the emoji visualizations and told the participants that they could press the disagree button at any point if they found the cognitive state prediction inaccurate. When participants finished the problem set, the fNIRS cap was removed and an informal interview was conducted. If the participant got hint prompts, we asked how they felt about them. We also asked if they found the cognitive state predictions accurate. The participants reviewed and signed an informed consent approved by WPI's IRB. Participants were compensated $15 for their time.

Participant	Prior Understanding of Material	Hint Prompt Shown	Comments	Notes
P101	Familiar, not confident	No	Did not notice any change in the emoji throughout the experiment	Error communicating with API, emoji and trigger sending was not functioning
P102	Familiar, not confident	Yes	"First hint prompt came up right as I was considering breaking the problem into step, second time it happened I declined and wanted to figure it out myself." Agreed with predictions for the most part, even during introduction problem when the experiment was being explained	N/A
P103	Familiar, not confident	Yes	"It said I was confused most of the time but I didn't necesarily agree, I got hint prompts when I didn't feel I needed them. I got a hint seconds after a problem started once"	N/A
P104	Completely unfamiliar	Yes	"Hints happened very frequently, there were times where I got hint prompts when I was not necessarily confused, but still thinking about the problem. In one instance I got a hint prompt right as the problem started"	N/A
P105	Familiar, somewhat	No	Generally found the emojis to be accurate, and did not find it distracting. Did not get any hint prompts, but did click the button himself at times. At one point tried to control the emoji himself by distracting himself when he understood a problem, the emoji changed accordingly. Didn't feel like he needed help, for most of the problems except for the second, for which he broke the problem into steps himself.	Hint Prompt Buffer now resets after each problem

Table 2: Table of feedback from the five participants of our pilot studies testing the hint system and visualizations.

Four of our five participants reported limited familiarity with probabilities while one of them (P105) was somewhat more confident, having had taken a probabilities course more recently. Participant P101 reported at the end of the session that the emoji had not changed at all in the duration of the study. This turned out to be a communication issue with the Chrome Extension which was resolved for the following participants. Three of the other participants (P102, P104, and P105) reported that they generally agreed with the cognitive state predictions displayed to them. P105 even claimed they were able to control the emoji consciously by intentionally distracting themselves and observing the change. The third participant (P103) reported that the emoji was confused for a majority of the problem set even in times where they felt confident in the answer. P102, P103, and P104 all reported that sometimes they got hint prompts when they did not need a hint, claiming that they were still thinking about the problem but not necessarily confused. P103 and P104 reported getting hint prompts moments after they began a problem, which was distracting for them since they did not have a chance to think about the problem before being prompted for a hint. We realized this was a design oversight on our part, the buffer of stored cognitive states of the user does not reset between problems in the problem set. This means that rule acquisition states from the previous problem affect the systems decision to show hints for the current problem. We corrected the issue by programming the buffer to clear when a new problem is started. The revised code was used with P105. They reported receiving no hint prompts during the problem set. During the unstructured interview, they expressed that they did not feel they needed assistance for the problem set, except for the second problem, where they broke the problem into steps before the 90 second time buffer for the hint prompt.

4.3.2 BrainBridge: Researcher

The researcher version of BrianBridge is catered toward research and analysis of the brain data as students work through problem sets. This version strips the extension of the visualizations and interventions that may distract students. It contains the CSV logging feature and the student question prompt to get more data on the students cognitive state.

Node.js as an installation requirement for this version of BrainBridge if the user is looking to log user actions to CSV files. This is because a Node.js server is required to access and save data to the users local hard drive.

The researchers at WPI and University of Pittsburg have conducted three pilot studies using the browser extension and have provided valuable feedback and feature requests. We helped develop the protocol of the pilot studies. We also asked them to give feedback on documentation of the browser extension and API to make sure they could build and operate the extension without additional help.

5 Discussion

In this study we demonstrated the ability to use real-time communication between brain data and an ITS to create interventions for students while learning. We also demonstrated the potential of these tools to help researchers better understand a student's cognitive state while learning.

5.1 User Studies

Our pilot studies, though limited in scope, show promise in BrianBridge's ability to create interventions and assess classifications of brain data. Many of the participants reported that the browser overestimated their confusion, which correlates with a known bias towards the rule acquisition state in the machine learning model. None of the participants expressed that the visualization was distracting. This may be a good indicator that the visualization was successful in avoiding the Midas touch effect by altering the users cognitive state. However, we noticed that none of the users clicked the disagree button, even though users reported disagreeing with predictions after the session. We predict that having to click the browser icon to reach the button made it too far out of the way for users who are focusing on completing the problem set. It is good that users did not feel distracted by the extension, but for research that values this feedback from the user a more overt system to report disagreements in cognitive state predictions may be needed. Users seemed to have more mixed reactions to the hint system. Participants P102, P103, and P104 all reported getting hint prompts when they did not feel they needed them, sometimes moments after the problem started.Resetting the buffer of stored cognitive states appeared to help remedy the problem, as P105 did report receiving any unnecessary hints. Adjusting the time buffer and rule acquisition percentage threshold such that hints were less frequent might have helped as well, but testing the effectiveness of different parameters was beyond the scope of this experiment.

The pilot studies also revealed some technical faults in the browser extension which limit the conclusions we can draw from them. When looking at the trigger logs it was evident that some triggers were not sending properly. For example, we noticed that some submit triggers were missing in the data despite it being impossible to proceed in the problem set without submitting problems. These observations made it possible to debug the software to ensure a smooth experience for future users, however, it means we cannot draw many conclusions from the triggers sent during the pilot studies.

5.2 Platform Flexibility

While the browser extension that we created was designed with the ASSISTments ITS in mind, this project can provide a framework for enabling BCI communication with other online platforms as well. The browser extension can read and visualize brain data anywhere in the browser, meaning the visualizations can be used for any web application. However, the features that require pulling data and interacting with the web page were designed specifically for the ASSISTments. This means that to read user actions or create interventions for different web applications, new code would need to written to read the web pages and interact with them. Trigger mapping also relies on the unique IDs assigned to ASSISTments problems, an equivalently unique identifier is also required to be accessible through the HTML DOM of the web application to create problem specific triggers.

Researchers can also expand the API to support models that can predict different cognitive states, or visualize raw brain data. This would entail the creation of new API calls that process and return the brain data in the desired format. Since the API communicates via an LSL stream, it can also receive signals from sensors and neuroimaging devices beyond fNIRS.

5.3 Future Work

This project is meant to provide a framework for communication between the brain and browser based ITS. More work needs to be done to get robust results on how visualizations and interventions affect students. Different visualizations affect different people differently [21]. Therefore it is important to create new visualizations and assess their effectiveness on students as well. Our work focused on the needs of our student and researcher personas. With more time we could have developed features to suit the needs of our teacher persona, Amy. A teacher such as Amy would benefit from understanding the cognitive state of their students. However, the browser extension does not store or log the cognitive states of the students in an easy to understand way. Amy is unfamiliar with how to interpret or process brain data in Aurora. A way to save the users cognitive state data, perhaps by expanding upon the CSV logging feature, and load that data into an easy to understand dashboard could be useful to Amy. The dashboard could highlight problems where students tended to be in the confused state more often, indicating where students may lack understanding.

6 Conclusion

The goal of this project was to bridge the gap between brain data and ITSs. The browser extension we created allows us create real-time visualizations and meaningful interventions based on the cognitive state of the user. Its also enables researchers to accurately map brain data to user actions in an ITS.

Our persona studies and feedback from researchers helped us determine what features were necessary to create a useful tool for our target users. We also designed an experiment around our own visualizations and interventions in the form of an automated hint-system based on the users predicted confusion during a problem set. The feedback from users in pilot studies illustrates the browser extensions ability to create new experiences for students using brain data. The action logging and trigger mapping features have also proven useful to researchers looking to better understand the cognitive states of the students during learning. While the browser extension was developed with the ASSISTments ITS in mind, it can be built upon to support more platforms and create a variety of experiences integrating BCI into web-based applications. We hope that our work is a step forward towards providing students with personalized online educational experiences powered by BCI.

References

[1] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, pp. 164–171, 2006.

[2] F. R. Mashrur, K. M. Rahman, M. T. I. Miya, R. Vaidyanathan, S. F. Anwar, F. Sarker, and K. A. Mamun, "Bci-based consumers' choice prediction from eeg signals: An intelligent neuromarketing framework," *Frontiers in Human Neuroscience*, p. 311, 2022.

[3] I. Martišius and R. Damaševičius, "A prototype ssvep based real time bci gaming system," *Computational intelligence and neuroscience*, vol. 2016, 2016.

[4] A. Trifoni and M. Shahini, "How does exam anxiety affect the performance of university students," *Mediterranean journal of social sciences*, vol. 2, no. 2, pp. 93–100, 2011.

[5] B. K. Sato, C. F. Hill, and S. M. Lo, "Testing the test: are exams measuring understanding?," *Biochemistry and Molecular Biology Education*, vol. 47, no. 3, pp. 296–302, 2019.

[6] J. Katona and A. Kovari, "Examining the learning efficiency by a brain-computer interface system," *Acta Polytechnica Hungarica*, vol. 15, no. 3, pp. 251–280, 2018.

[7] C. Crescentini, S. Seyed-Allaei, N. De Pisapia, J. Jovicich, D. Amati, and T. Shallice, "Mechanisms of rule acquisition and rule following in inductive reasoning," *Journal of Neuroscience*, vol. 31, no. 21, pp. 7763–7774, 2011.

[8] M. Ferrari and V. Quaresima, "A brief review on the history of human functional near-infrared spectroscopy (fnirs) development and fields of application," *NeuroImage*, vol. 63, no. 2, pp. 921–935, 2012.

[9] N. Naseer and K.-S. Hong, "fnirs-based brain-computer interfaces: a review," *Frontiers in human neuroscience*, vol. 9, p. 3, 2015.

[10] E. T. Solovey, A. Girouard, K. Chauncey, L. M. Hirshfield, A. Sassaroli, F. Zheng, S. Fantini, and R. J. Jacob, "Using fnirs brain sensing in realistic hci settings: Experiments and guidelines," in *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, UIST '09, (New York, NY, USA), p. 157–166, Association for Computing Machinery, 2009.

[11] P. Pinti, I. Tachtsidis, A. Hamilton, J. Hirsch, C. Aichelburg, S. Gilbert, and P. W. Burgess, "The present and future use of functional near-infrared spectroscopy (fnirs) for cognitive neuroscience," *Annals of the New York Academy of Sciences*, vol. 1464, no. 1, pp. 5–29, 2020.

[12] S. K. Mudgal, S. K. Sharma, J. Chaturvedi, and A. Sharma, "Brain computer interface advancement in neurosciences: Applications and issues," *Interdisciplinary Neurosurgery*, vol. 20, p. 100694, 2020.

[13] G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy, "A high-performance brain–computer interface," *nature*, vol. 442, no. 7099, pp. 195–198, 2006.

[14] M. J. Khan and K.-S. Hong, "Passive bci based on drowsiness detection: an fnirs study," *Biomed. Opt. Express*, vol. 6, pp. 4063–4078, Oct 2015.

[15] S. F. Verkijika and L. De Wet, "Using a brain-computer interface (bci) in reducing math anxiety: Evidence from south africa," *Computers & Education*, vol. 81, pp. 113–122, 2015.

[16] K. R. Koedinger, A. T. Corbett, and C. Perfetti, "The knowledge-learning-instruction (kli) framework: Toward bridging the science-practice chasm to enhance robust student learning," *Cognitive Science*, 2010.

[17] T. S. Braver, A. Kizhner, R. Tang, M. C. Freund, and J. A. Etzel, "The dual mechanisms of cognitive control project," *Journal of Cognitive Neuroscience*, vol. 33, no. 9, pp. 1990–2015, 2021.

[18] J. Nielsen, "Ten usability heuristics," 2005.

[19] E. T. Solovey, D. Afergan, E. M. Peck, S. W. Hincks, and R. J. Jacob, "Designing implicit interfaces for physiological computing: Guidelines and lessons learned using fnirs," *ACM Transactions on Computer-Human Interaction*, vol. 21, 2015.

[20] T. O. Zander, C. Kothe, S. Jatzev, and M. Gaertner, "Enhancing human-computer interaction with input from active and passive brain-computer interfaces," 2010.

[21] E. M. M. Peck, B. F. Yuksel, A. Ottley, R. J. Jacob, and R. Chang, "Using fnirs brain sensing to evaluate information visualization interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 473–482, 2013.

[22] K. VanLehn, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," *Educational psychologist*, vol. 46, no. 4, pp. 197–221, 2011.

[23] V. Aleven and K. R. Koedinger, "Limitations of student control: Do students know when they need help?," in *Intelligent Tutoring Systems: 5th International Conference, ITS 2000 Montréal, Canada, June 19–23, 2000 Proceedings 5*, pp. 292–303, Springer, 2000.

[24] A. Sarrafzadeh, S. Alexander, F. Dadgostar, C. Fan, and A. Bigdeli, ""how do you know that i don't understand?" a look at the future of intelligent tutoring systems," *Computers in Human Behavior*, vol. 24, no. 4, pp. 1342–1363, 2008.

[25] N. T. Heffernan and C. L. Heffernan, "The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching," *International Journal of Artificial Intelligence in Education*, vol. 24, pp. 470–497, 2014.

[26] P. Kehrer, K. Kelly, and N. Heffernan, "Does immediate feedback while doing homework improve learning?.," *Grantee Submission*, 2013.

[27] K. Kelly, N. Heffernan, C. Heffernan, S. Goldman, J. Pellegrino, and D. Soffer Goldstein, "Estimating the effect of web-based homework," in *Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013. Proceedings 16*, pp. 824–827, Springer, 2013.

[28] R. S. Kim, R. Weitz, N. T. Heffernan, and N. Krach, "Tutored problem solving vs."pure" worked examples," in *Proceedings of the 31st annual conference of the cognitive science society*, pp. 3121–3126, Cognitive Science Society Austin, TX, 2009.

[29] A. Chidambaranathan, H. Krishnamoorthy, C. N. S. P. M. Chandu, S. Khandelwal, and S. Harikumar, "Learning from youtube videos using drona extension," pp. 1–6, IEEE, 7 2021.

[30] M. W. Ok and K. Rao, "Digital tools for the inclusive classroom: Google chrome as assistive and instructional technology," *Journal of Special Education Technology*, vol. 34, pp. 204–211, 9 2019.

[31] V. Martinez-Cagigal, J. Gomez-Pilar, D. Alvarez, and R. Hornero, "An asynchronous p300-based brain-computer interface web browser for severely disabled people," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, pp. 1332–1342, 8 2017.

[32] "Browser market share worldwide."

[33] A. Cooper, "The inmates are running the asylum," in *Software-Ergonomie'99*, pp. 17–17, Springer, 1999.

[34]

[35] Pallets, "Welcome to flask — flask documentation (2.2.x)." https://flask.palletsprojects.com/en/2.2.x/.

[36] Mozilla, "Mutation observer - web apis — mdn." https://developer.mozilla.org/en-US/docs/Web/API/MutationObserver.

[37] M. Prada, D. L. Rodrigues, M. V. Garrido, D. Lopes, B. Cavalheiro, and R. Gaspar, “Motives, frequency and attitudes toward emoji and emoticon use,” *Telematics and Informatics*, vol. 35, no. 7, pp. 1925–1934, 2018.

[38] A. Howell-Munson, T. Moward, D. S. Unal, C. Arrington, and E. T. Solovey, “Classification of brain signals collected during a rule learning paradigm,” *Artificial Intelligence in Education: 24th International Conference, AIED 2023, Tokyo, Japan, July 3–7, 2023, Proceedings*, 2023.

Appendices

A GitHub Repositories

The front end code for BrainBridge can be found here: https://github.com/DavidD121/BCI-Chrome-Extension/ The main branch represents BrainBridge: Student See the ”NCS” Branch for BrainBridge Researcher

The API code can be found in the Neurolearn repository in the API folder: https://github.com/WPIHCILab/NeuroLearn

See the README files in each repository for build instructions

B API Schema

See the api-reference.md file in the Nuerolearn Github Repository in Appendix A for a better formatted markdown file

B.1 /ProblemStart

B.1.1 Description

Send message to indicate that a new problem has started, sending a trigger to Aurora if the problem is mapped to one.

B.1.2 Type

- POST

B.1.3 Request:

`json5 { "problem_id": string, /* Current problem ID */ }`

B.1.4 Response

- 200 - Success
- 400 - Failure

B.2 /ProblemSubmit

B.2.1 Description

Send message to indicate that a problem has been submitted, noting correctness, sending a trigger to Aurora if the problem is mapped to one.

B.2.2 Type

- POST

B.2.3 Request:

`{ "problem_id": string, /* Submitted problem ID */ "correct": bool /* Whether the submission was correct or incorrect */ }`

B.2.4 Response

- 200 - Success
- 400 - Failure

B.3 /Step

B.3.1 Description

Send message to indicate that a new step has started in a problem broken into setps, sending a trigger to Aurora if the problem is mapped to one.

B.3.2 Type

- POST

B.3.3 Request:

`json5 { "step_id": int, /* Current step ID */ }`

B.3.4 Response

- 200 - Success
- 400 - Failure

B.4 /GetRLState

B.4.1 Description

Gets the current prediction of the users current rule learning state, rule following or rule acquisition, based on readings from Aurora.

B.4.2 Type

- GET

B.4.3 Request:

N/A

B.4.4 Response

`json5 { "state": int /* Current predicted rule learning state. 0 for rule acquisition, 1 for rule following */ }`

B.5 /Disagree

B.5.1 Description

Send message to indicate that the user has disagreed with the cognitive state prediction.

B.5.2 Type

- POST

B.5.3 Request:

`json5 { "state": int /* Predicted rule learning state when the user disagreed. 0 for rule acquisition, 1 for rule following */ }`

B.5.4 Response

- 200 - Success
- 400 - Failure

B.6 /rnd

B.6.1 Description

Returns a random float between 0 and 1 with 2 decimal places Sends trigger to aurora. Meant for testing

B.6.2 Type

- GET

B.6.3 Request:

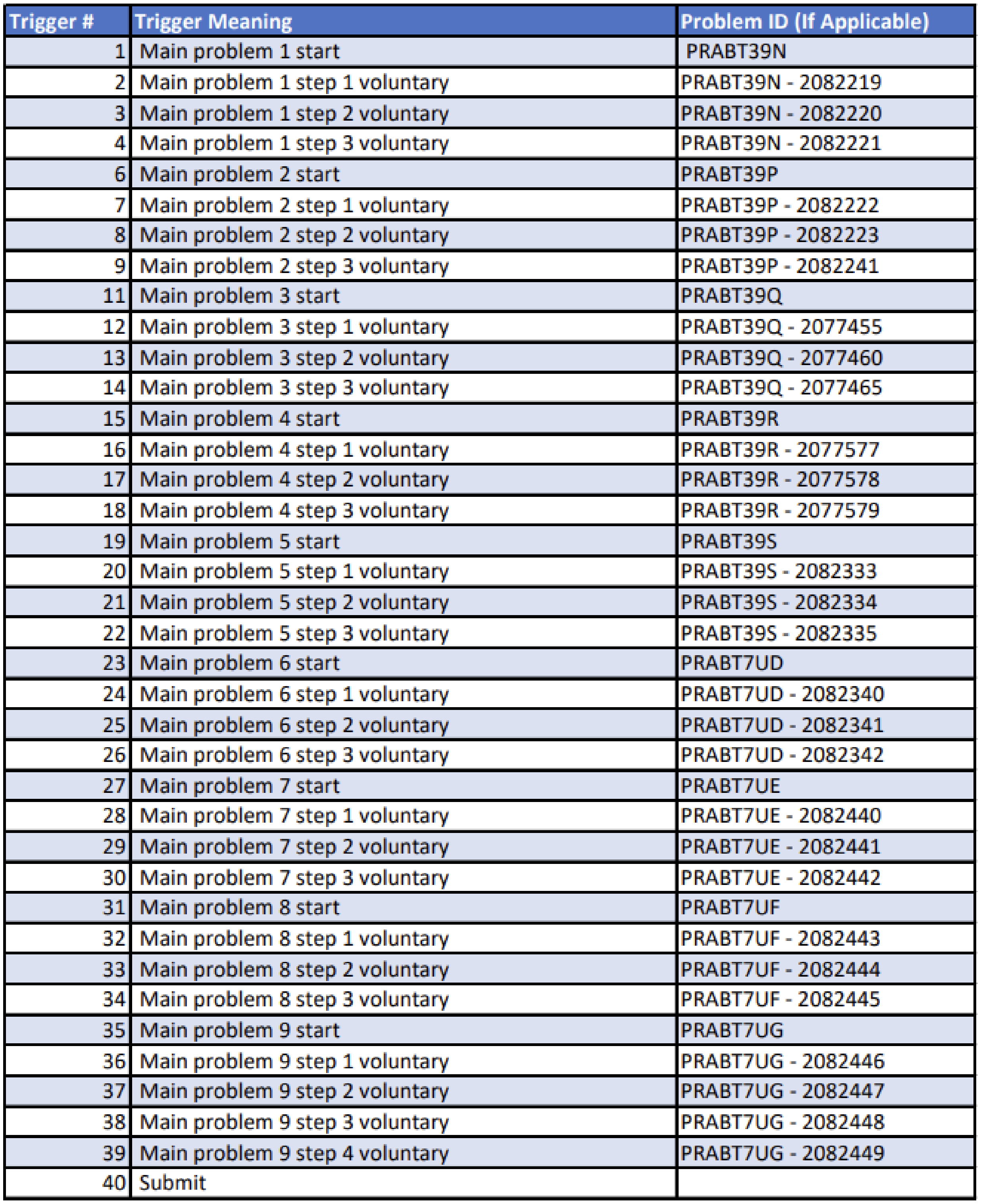
N/A

B.6.4 Response

‘‘‘json5 { ”state”: int /* Current predicted rule learning state. 0 for rule acquisition, 1 for rule following */ }

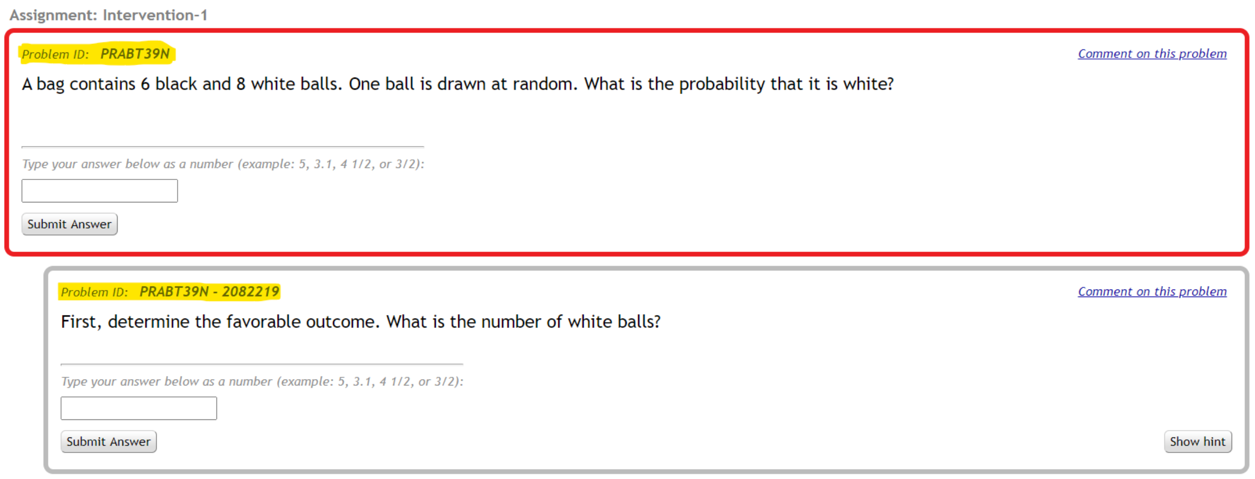
C Example of Trigger Mapping

The following is an outline of the Trigger Mapping for the problem set used during development.



Trigger #	Trigger Meaning	Problem ID (If Applicable)
1	Main problem 1 start	PRABT39N
2	Main problem 1 step 1 voluntary	PRABT39N - 2082219
3	Main problem 1 step 2 voluntary	PRABT39N - 2082220
4	Main problem 1 step 3 voluntary	PRABT39N - 2082221
6	Main problem 2 start	PRABT39P
7	Main problem 2 step 1 voluntary	PRABT39P - 2082222
8	Main problem 2 step 2 voluntary	PRABT39P - 2082223
9	Main problem 2 step 3 voluntary	PRABT39P - 2082241
11	Main problem 3 start	PRABT39Q
12	Main problem 3 step 1 voluntary	PRABT39Q - 2077455
13	Main problem 3 step 2 voluntary	PRABT39Q - 2077460
14	Main problem 3 step 3 voluntary	PRABT39Q - 2077465
15	Main problem 4 start	PRABT39R
16	Main problem 4 step 1 voluntary	PRABT39R - 2077577
17	Main problem 4 step 2 voluntary	PRABT39R - 2077578
18	Main problem 4 step 3 voluntary	PRABT39R - 2077579
19	Main problem 5 start	PRABT39S
20	Main problem 5 step 1 voluntary	PRABT39S - 2082333
21	Main problem 5 step 2 voluntary	PRABT39S - 2082334
22	Main problem 5 step 3 voluntary	PRABT39S - 2082335
23	Main problem 6 start	PRABT7UD
24	Main problem 6 step 1 voluntary	PRABT7UD - 2082340
25	Main problem 6 step 2 voluntary	PRABT7UD - 2082341
26	Main problem 6 step 3 voluntary	PRABT7UD - 2082342
27	Main problem 7 start	PRABT7UE
28	Main problem 7 step 1 voluntary	PRABT7UE - 2082440
29	Main problem 7 step 2 voluntary	PRABT7UE - 2082441
30	Main problem 7 step 3 voluntary	PRABT7UE - 2082442
31	Main problem 8 start	PRABT7UF
32	Main problem 8 step 1 voluntary	PRABT7UF - 2082443
33	Main problem 8 step 2 voluntary	PRABT7UF - 2082444
34	Main problem 8 step 3 voluntary	PRABT7UF - 2082445
35	Main problem 9 start	PRABT7UG
36	Main problem 9 step 1 voluntary	PRABT7UG - 2082446
37	Main problem 9 step 2 voluntary	PRABT7UG - 2082447
38	Main problem 9 step 3 voluntary	PRABT7UG - 2082448
39	Main problem 9 step 4 voluntary	PRABT7UG - 2082449
40	Submit	

The trigger number is the integer that is sent to Aurora, the trigger meaning is the reason the trigger was send, and if said trigger corresponds to an ASSISTments problem/step, the ID of said problem must be known as well. The IDs of ASSISTments problems can be found in the as you go through the problem sets, see below (IDs highlighted in yellow)


Assignment: Intervention-1

Problem ID: PRABT39N

A bag contains 6 black and 8 white balls. One ball is drawn at random. What is the probability that it is white?

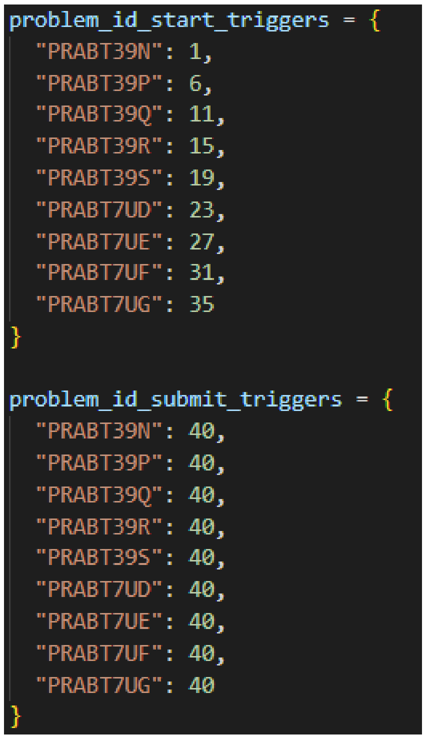
Type your answer below as a number (example: 5, 3.1, 4 1/2, or 3/2):

Problem ID: PRABT39N - 2082219

First, determine the favorable outcome. What is the number of white balls?

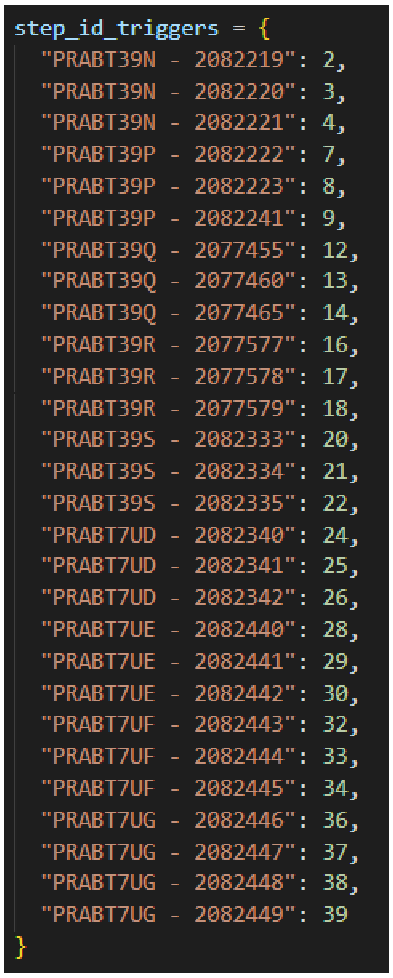
Type your answer below as a number (example: 5, 3.1, 4 1/2, or 3/2):

To apply the trigger mappings in code, navigate to the "trigger_mappings.py" file. Here you will find different Python dictionaries corresponding to different trigger types, one for starting problems, one for submitting problems, and one for entering a step. For each trigger assign the problem/step ID as the key, and the trigger number it corresponds to as the value. No triggers will be sent for problems which are not mapped to a trigger in this file. If a trigger is generic, such as the submit trigger in the provided example mapping, include all the problems for which it should be triggered in the dictionary. The defined dictionaries in "trigger_mapping.py" for the provided example mapping are shown below:



```
problem_id_start_triggers = {
  "PRABT39N": 1,
  "PRABT39P": 6,
  "PRABT39Q": 11,
  "PRABT39R": 15,
  "PRABT39S": 19,
  "PRABT7UD": 23,
  "PRABT7UE": 27,
  "PRABT7UF": 31,
  "PRABT7UG": 35
}

problem_id_submit_triggers = {
  "PRABT39N": 40,
  "PRABT39P": 40,
  "PRABT39Q": 40,
  "PRABT39R": 40,
  "PRABT39S": 40,
  "PRABT7UD": 40,
  "PRABT7UE": 40,
  "PRABT7UF": 40,
  "PRABT7UG": 40
}
```



```
step_id_triggers = {
  "PRABT39N - 2082219": 2,
  "PRABT39N - 2082220": 3,
  "PRABT39N - 2082221": 4,
  "PRABT39P - 2082222": 7,
  "PRABT39P - 2082223": 8,
  "PRABT39P - 2082241": 9,
  "PRABT39Q - 2077455": 12,
  "PRABT39Q - 2077460": 13,
  "PRABT39Q - 2077465": 14,
  "PRABT39R - 2077577": 16,
  "PRABT39R - 2077578": 17,
  "PRABT39R - 2077579": 18,
  "PRABT39S - 2082333": 20,
  "PRABT39S - 2082334": 21,
  "PRABT39S - 2082335": 22,
  "PRABT7UD - 2082340": 24,
  "PRABT7UD - 2082341": 25,
  "PRABT7UD - 2082342": 26,
  "PRABT7UE - 2082440": 28,
  "PRABT7UE - 2082441": 29,
  "PRABT7UE - 2082442": 30,
  "PRABT7UF - 2082443": 32,
  "PRABT7UF - 2082444": 33,
  "PRABT7UF - 2082445": 34,
  "PRABT7UG - 2082446": 36,
  "PRABT7UG - 2082447": 37,
  "PRABT7UG - 2082448": 38,
  "PRABT7UG - 2082449": 39
}
```