Developing a Machine Learning Algorithm to Assess Attention Levels in ADHD Students in a Virtual Learning Setting using Audio and Video Processing

Srivi Balaji, Meghana Gopannagari, Svanik Sharma, Preethi Rajgopal

Abstract: ***Over the past few years, numerous technological advancements have modernized and eased access to educational materials, improving overall learning experiences for students with ADHD despite the transition to remote learning. However, the majority of these improvements address comprehension and practice outside of the classroom without recognizing the need for engagement during a lesson. Students are more likely to retain higher amounts of information outside of class, if they have a strong understanding of the lesson during class. A back-end model combined with an engaging front-end user interface can enhance the standard of education for students with ADHD and help them achieve the same level of understanding they would have during an in-person lesson. This project aimed to address the remote learning experiences of students with ADHD by creating a model using machine learning to analyze audio and video clips of a live online lesson, detect distractions in the student's environment, and use this data in tandem with an interactive user interface to engage students and enhance their remote learning experience. The two means of data collection employed in this model were audio and video analysis. This data was fed into separate convolutional neural networks with reinforcement learning architecture to identify distractions. A genetic algorithm was used to weigh the outputs of both neural networks and produce coefficients determining the weight of each factor. This was then used to determine the distraction level of the student. This model can be implemented in a virtual lesson between an instructor and a student with ADHD, to constantly monitor the attention level of the student. Findings of this research suggested that this model could help an instructor acknowledge and manage symptoms of ADHD – which may lead to distractions, such as impulsivity, hyperactivity and boredom – by modifying their curriculum to further engage the student. This research has the potential to fill the notable gap between technology and education, using technology to improve online educational quality for students with ADHD.***

Keywords: ADHD, genetic algorithm, machine learning, neural networks, virtual education

Manuscript received on May 21, 2021.
Revised Manuscript received on May 28, 2021.
Manuscript published on May 30, 2021.

* Correspondence Author
Srivi Balaji, Meridian World School, Round Rock, United States of America. Email: srivibalaji33@gmail.com
Meghana Gopannagari, Thomas Jefferson School of Science and Technology, Alexandria, United States of America. Email: meghana.gopannagari@gmail.com
Svanik Sharma, Vandegrift High School, Leander, United States of America. Email: ssharma10393@gmail.com
Preethi Rajgopal, Kelley School of Business, Indiana University - Bloomington, Bloomington, United States of America. Email: prajgopa@iu.edu

I. INTRODUCTION

Due to the CoronaVirus Pandemic, students are required to participate in classes virtually. However, because of the reduction of academic support students have from their teachers during virtual learning, it is increasingly difficult for students to learn virtually. Students struggle to pay attention in class virtually, and for students with ADHD, or Attention Deficit Hyperactivity Disorder, staying focused in class is even more difficult. The paper outlines a tool for teachers to better understand the attention levels of their class and to encourage their students to stay attentive in class. This research was conducted by exploring different algorithms that use audio and video processing to determine the attention rates of an individual. To ensure that this tool will help students with ADHD, the algorithms follow the guidelines for designing software for ADHD students proposed by the scientific community. This paper will outline the methods used to develop an algorithm that outputs the attention levels of students and design an engaging user interface that allows students to learn efficiently and teachers to teach effectively.

II. LITERATURE REVIEW

This paper addresses the issue of remote learning experiences for students with ADHD by developing a machine learning model to take audio and video clips of a student during live online instruction, analyze the data to identity distractions and a lack of attention from the student, and use this data in conjunction with a user interface designed for students with ADHD, to provide an engaging, interactive learning experience.

This project is targeted to support students with Attention Deficit Hyperactivity Disorder (ADHD). ADHD is a disorder that deals with mental health, causing impulsive decisions and excessive hyperactivity. Students with ADHD may struggle to focus on a task or lesson, and they may have trouble sitting still for long times. This especially negatively impacts them during school [3]. In order to address symptoms of ADHD, the model described in this paper uses machine learning, a subset of artificial intelligence (AI).

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication

Machine learning enables computer programs to use a set of data to "learn" to perform a function, whether that be classification, regression, density estimation, etc. This way, the computer program does not have explicit classifier instructions, but learns from the data and its previous evaluations to develop an accurate model [37]. Although there are pre-existing algorithms that have been developed for various purposes (data analysis, image recognition, trend predictions, etc.), individual models often must be developed using features from various algorithms to serve a specific purpose [19]. For example, a simple model to predict future trends on the stock market may require the use of a regression algorithm to establish a relationship between an input and its corresponding output, as well as a prediction algorithm to use this relationship and determine the most likely output for untested inputs.

In this way, this paper will explore the development of a model through various machine learning algorithms for data analysis (audio and video) as well as a genetic algorithm to use the data to determine the distraction level of a student with ADHD during a virtual lesson. This model will be programmed in tandem with the development of a user interface (UI) designed specifically for students with ADHD. The UI enables humans to interact with a computer through device features such as keyboards, displays and mouses. In this case, the UI display screen will be developed based on the data from the model, to provide an engaging educational experience [14].

A. Education

Over the past decade, countless technological inventions have contributed to the advancement and modernization of educational material and learning experiences. Approximately 75% of educational instructors envision the replacement of physical textbooks with virtual material by 2026 [5]. Multiple platforms of education software and online learning programs allows students to independently further their education and continue learning based on their current comprehension level. In light of the COVID-19 pandemic, in the spring of 2020, schools across the country switched from in-person to remote learning from home. For the past year, platforms such as Zoom and Microsoft Teams have come under the public spotlight as the primary distance schooling platforms. These have allowed students to continue at roughly the same level of comprehension after the transition to distance learning [36].

However, symptoms of ADHD such as hyperactivity and limited attention span have impacted students' ability to adapt to the sudden virtual change [6]. Students especially struggle with hyperactivity as they sit for numerous hours at a time in front of the computer. Hyperactivity is generally regulated by physical activity, but this has been limited by remote learning. This also causes an increase in boredom and a decrease in engagement as well as motivation to tackle difficult problems. Online learning has also limited social relationships and free communication with other students, which is imperative for children with ADHD. This furthers their challenge to develop and maintain social relationships, and increases isolation and anxiety during lessons [6]. Students with ADHD often have other disorders (such as learning disorders or behavioral problems) which makes it even more difficult to cope with change and social isolation.

Numerous technological advancements have been made to improve the quality of education for students with ADHD. Smartphones have allowed students with ADHD to develop their working memory and use pictures or files to help them access homework and any material. Smart pens such as Livescribe and Neo Smartpen also allow students with ADHD to take notes with parallel audio recordings that they can then access at a later time. This allows students to have the verbal instruction and their notes anytime they need to retrieve this information. Of course, hundreds of apps exist to help students with handwriting, working memory and weak cognitive functions [34]. Although these developments have notably improved the learning rates of students with ADHD, very few of these tackle the root problem: engagement and concept comprehension during a lesson [27]. By improving a student's understanding of the lessons and concepts taught during a class, they are more likely to retain the understanding outside of class. While technology has greatly advanced the availability of information for the general public, there is a digital learning gap that limits the full use of technology for the highest quality of education. Therefore, this paper strives to address this gap and develop a model that can use audio and video analysis to detect distractions in a student's environment during virtual instruction, using this to create an engaging user interface that appeals to students with ADHD and enhances their overall online learning experience.

B. Technology

Machine Learning Algorithms: Supervised, Unsupervised and Reinforcement Learning

Algorithms describe the process for finding the solution to a specific type of problem. It is a compilation of a set of steps, formulas or actions to be followed the exact same way each time the algorithm is run – given an input or set of inputs, the algorithm will perform the same steps to derive an output or set of outputs. For example, a recipe for a certain food would be an example of an algorithm: the recipe provides a detailed account of steps to make the food from start to finish. The recipe will take a set of inputs (ingredients) and an individual may perform the steps to derive the output (food product).

When evaluating the level of distraction from various audio and video signals, 3 types of machine learning algorithms were in candidacy: supervised learning, unsupervised learning, and reinforcement learning. Almost all machine learning algorithms that use data analysis (e.g. video, audio, images, etc.) are categorized into one of these three sections. Supervised learning employs a variable output which can be determined through a group of predictors or variable inputs.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

These algorithms require 2 pre-established datasets: a training and testing dataset. Using these datasets and the predictors, the model is able to map inputs to the predicted output, increasing its accuracy each time it is evaluated [26]. Supervised learning can be used commonly for regression (continuous output for each input) and classification (predicts the label of the input). Within this set of algorithms, the purpose of the predictors is to establish a correlation between known input and output values, in order to predict output values for unknown input values with the highest accuracy. Supervised learning models are trained to identify this correlation with the training dataset (known input and output), and test its performance with the test dataset of known input and unknown output [10]. Although supervised learning would be extremely effective when analyzing data based on precedent [17], the classifier requires an extensive amount of data in order to work best. In addition, any incorrect or unclear labels in the training dataset can negatively impact the accuracy of the model. Larger numbers of classification labels require larger amounts of data, and will take an extensive amount of computing time [17]. For example, 5000 frames or images may take up to a few minutes to pre-classify the input data – however, the accuracy of the model when testing may only be around 0.7 or 70%. Increasing the accuracy of the model may necessitate a drastic increase of input data. Tens or hundreds of thousands of images may have a much higher accuracy rate, although evaluating every input can take a number of hours.

In contrast, with unsupervised learning, there is not a set of labels of classifiers for the model to classify into. The unsupervised learning algorithm is used to generically cluster groups of data based on patterns and similarities. Therefore, two sets of data fed into the model may not output the same set of classifiers [26]. This model is generally used for density estimation and clustering, as it can describe patterns in the data without needed pre-existing labels. It can be used to identify the general structure of a dataset whose trends cannot realistically be identified manually – in situations like this, unsupervised learning can be used to identify patterns or correlations in the data, which can then be used to test individual hypotheses about the data [10]. However, it is important to note that unlike the supervised learning model, it is more difficult to evaluate the performance of the model, as there is no test dataset with the "correct" labels. More evaluations are also necessary to reach a high level of accuracy, as the input data is not known and cannot be used to provide information for the model. In order to avoid this, the input data for every dataset should be classified beforehand to evaluate the accuracy of the model, which can be, as explained before, quite time consuming [18]. In the context of a virtual lesson, time is an important factor as increased computation time will cause a greater delay of outputting the distraction level for a student, which is not favorable in an educational environment (educators need "real time" information about the distraction level of their students immediately.)

Finally, reinforcement learning describes the process through which a programming agent uses trial-and-error interactions to *learn* behaviors in a dynamic environment [16]. This way, the agent is rewarded or punished based on how successfully it achieves the task at hand. This sort of human-like training of **computer object abstractions** (the agents that are learning the task) has increased its importance and applications in the AI and machine learning fields over the past decade [16]. Based on the algorithms used, reinforcement learning problems can be solved in two ways. One strategy is to search through the entire space of behaviors to identify the one with the best performance in the environment. This method is most commonly used in problems involving genetic programming and other similar search approaches. The second technique, however, is focused on calculating the utility of taking individual actions within the environment, based on statistical analysis and dynamic programming. This method cannot be used for all optimization problems, as most do not have the problem structure required for this technique [16].

Delving into the organization of a general reinforcement learning model, an agent is able to observe its environment through action and perception. There are 3 key parts of the model: the set of states the environment can be in (S), the set of actions the agent can take (A), and the output reinforcement signals in the form {0,1} [9]. The agent receives the environment's current state, or the identity function, as input – *i*. The output is the action (*a*) that the agent wishes to use based on its input. Fig. 1 provides a visual representation of this model. A **scalar reinforcement signal** is used for the agent to receive the value of the new environment based on the action's change to the environment. Positive values correspond with rewards, and negative values correspond with punishments. As the values add up, they contribute to the long-term reinforcement measure. By repeating this process of trial-and-error throughout the duration of the program (directed by the algorithms used), the agent is able to best identify actions for specific states that will lead to higher scalar reinforcement signal values [16]. This sort of model assumes unpredictability when faced with a state; in other words, when an agent is faced with a state and takes the same action twice, the outcomes may be different. The probability of the action changing the state to receive a value, stays the same [4].

There are quite a few differences between reinforcement learning and supervised learning, which is more commonly used. Whereas supervised learning has a set of input and output pairs to train the computer model before testing it, in reinforcement learning, the agent's only indication of an optimal output is through the rewards system.

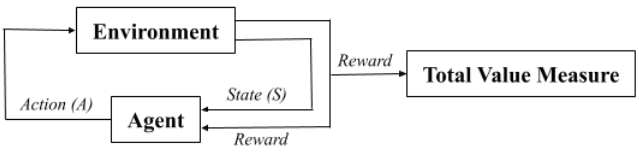

Figure 1: Basic Reinforcement Learning Model

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Therefore in reinforcement learning, the agent has no way of knowing which output would have been the best one for the highest reward; it simply needs to connect the states, actions and rewards in the best way to have an **optimized performance**. In addition, while the training stage of supervised learning is independent from the testing stage, these are combined in reinforcement learning; as the machine tests itself, it continuously learns from its rewards and punishments [16].

Reinforcement learning can be used in tandem with neural networks, which is useful for data classification based on spatial information [1]. Neural networks operate similar to neurons in the brain. They can identify complex relationships in data and use that to predict its outcome [8]. For example, in a dataset of cats and dogs, a neural network can be employed to identify the distinguishing characteristics between dogs and cats such as size, whiskers, appearance, etc. and use this to accurately classify an unknown image as a cat or dog.

Therefore, virtual agents will use convolutional neural networks with reinforcement learning architecture – in other words, the model combines **target optimization** with **function approximation** – to link environment states and actions for the highest reward [35]. The following section will introduce the use of convolutional neural networks in parallel with reinforcement learning to identify distractions based on audio and video analysis. For the purposes of this model – to analyze audio and video clips, identifying points of distraction or attention span – the best candidate was **reinforcement learning**.

When analysing audio, it is best to use a programming language with numerous uses in **machine learning**. Languages like Python, R, C++, Java and JavaScript would be suitable for audio processing [11]. Various machine learning algorithms and models have been developed for the sole purpose of audio analysis and speech processing, such as pyAudioProcessing, pyAudioAnalysis, hmmlearn and sklearn [23]. In order to develop these models, data analysts use different audio processing libraries, each with their own purpose. Specifically in Python, multiple libraries read audio such as pydub, pyAudioAnalysis, SciPy and libROSA [23]. There are also libraries for Audio Digital Signal Processing in other programming languages, including Q in C++, TarsosDSP in Java and SoundJS in JavaScript. Audio data must be retrieved in a high-resolution format, as well as one that is computer readable. Some of these formats include WMA (Windows Media Audio), mp3 (MPEG-1 Audio Layer 3), and wav or Waveform Audio File [32]. It is encouraged to have a **sampling rate** of 8kHz (sampling interval of 1/8 of a second), which is enough for regular audios. For better quality, a sampling rate of 44.1kHz (sampling interval of 1/44.1 of a second) can also be used [28]. This sampling rate can be achieved on a local computer.

Audio Analysis

Audio processing is an essential part of developing an engaging experience, as it can provide insight into the student's attention span, background distractions, response time through speech processing, and correlations between music and performance [29]. An essential application of audio processing is related to **background noise**. Measuring the student's background noise level can offer insight into any distractions that may prohibit the student from engaging with the lesson. In addition, any audio identification of fidgeting noises, such as a constant tapping in the background, may indicate a lack of attention or interest from the student. Speech on the student's part may be processed by measuring the response time of the individual after the instructor has posed a question; an increase in response time may indicate reduced lesson comprehension and/or engagement to a smaller extent. On the other hand, **speech processing** may also be used for the instructor, to gauge the speaking rate in comparison with that of the student; slower speaking rates may allow for a higher level of understanding, depending on the individual. With regards to **optimal background frequency** (or range of frequencies), music with a limited frequency range may be introduced as an independent variable which will affect the student's performance as measured through the resulting quantitative and qualitative data [33]. A specific range of frequencies may prove to have the highest positive impact on the student's learning experience. However, this should be experimented in a simulation setting in order to reduce any negative impact on a student's performance when implemented in a real life experience.

2 types of graphs will most commonly be used for audio processing: the time-domain graph and the frequency-domain graph. **The time-domain graph** is used to represent a signal's change over a period of time. It will be primarily used to identify signal patterns over time, as well as investigate the response time of a student when posed with a question [31]. This type of graph is useful for analysis that requires numerical measurements (in seconds), which constitutes a great part of audio processing for the learning model described in this paper. Time-domain analysis is used to measure background noise by gauging the level of noise in the background over a set amount of time. Any patterns in the signal can also be deciphered to indicate repetitive background noise or fidgeting noises, i.e. continuous tapping on a table. The student's response time can be tracked by calculating the time between the instructor posing a question, and the student answering it. Finally, by analysing individual words (which may require a higher sampling rate), time-domain graphs may

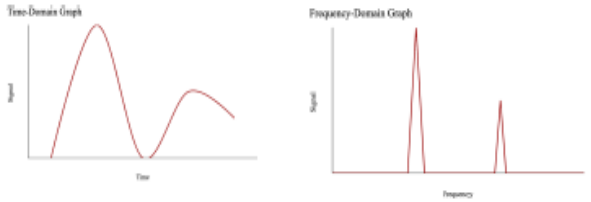

Figure 2: Visual comparison between time-domain graphs and frequency-domain graphs

be used to calculate the instructor's and student's speaking times by calculating the amount of individual words spoken over a period of time, i.e. 60 seconds [34].

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication

For analysis that references frequency rather than time, a **frequency-domain graph** may be employed. Frequency-domain compares a range of frequencies to the proportion of signal found within each band. These graphs analyze mathematical functions or signals using frequency rather than time. Therefore, instead of measuring in seconds, any relevant calculations will be in Hertz. **Frequency domain graphs** can be used to identify the optimal frequency to improve a student's learning experience. The primary way frequency-domain is implemented in this project, is through the introduction of music. The frequency domain graph will measure performance improvement, if any, with the addition of music in the background; frequency-domain can provide a more comprehensive analysis by breaking down the music signals into intricate components and measuring the separate frequencies present in each component. This data will assist in investigating the best range of frequencies for improving the performance level of a student with ADHD.

The general difference between a time-domain and frequency-domain graph can be seen in Fig. 2. When converting the signals from the time domain to frequency domain for musical analysis, the most common transform used is the Fourier transform. This breaks the time function down into frequencies by taking the integrals of each sine-wave present from the signal. The compilation of these frequencies makes up the frequency-domain graph. Although it will likely not be used in this project, the inverse Fourier transform can be used to reverse the process to turn a frequency-domain graph into time-domain [21].

Video Analysis

The two means of data collection employed in this model are audio and video analysis, to improve the learning experiences of students with ADHD. **Video analysis** is especially critical to ensuring that students are engaging with their environment. This section explores how video analysis, **eye vergence,** and **binocular vision** in ADHD children can be processed by a machine learning model to determine whether a subject is paying attention or not. ADHD students are particularly prone to **convergence insufficiency**, a common vision disorder characterized by double vision, blurred vision, eye strain, and difficulty concentrating [39]. While ethical controversy exists around the usage of eye-tracking technology for monitoring students because of its tendency to discriminate against students of different races and its improper usage by professors and teachers, this model is intended to be applied to those with attention disorders, specifically. Thus, this model is not anticipated in being used to monitor student behaviour.

Recent research into **eye-tracking algorithms** shows that **convolutional neural networks** can be used to track eye-movement in individuals. While infrared eye-tracking and electro-oculography have been researched more heavily, they require special instruments and equipment. Therefore, though less-researched, convolutional neural networks offer a less invasive and low cost method to track eye-movement. By tracking eye-movement and calculating the angle of vergence in ADHD students, it is possible to determine whether a student is paying attention to their coursework. In particular, the **task demand** of a student can be measured, which refers to the amount of cognitive resources used to complete a particular task. Previous research in eye-tracking and task demand demonstrates that other ocular data can be processed to measure the attention and cognitive effort of a person on a particular task, including fixation, saccade, blink, and pupillary response. **Fixation** refers to relatively stable gaze points that are near in both space and time, i.e, the tendency to focus on an object. Measuring fixation can allow us to determine what particular object or task the student is fixating on. **Saccades** are small, rapid eye movements when jumping from fixating on one object to another. These eye movements can indicate how long an individual is concentrating on an object of interest (such as the camera or a particular region of their computer screen) and is particularly varied in individuals with ADHD compared to the rest of the population. **Blink** refers to the involuntary shutting and opening of one's eye, and is known to be an indicator of cognitive attention. **Pupillary response** refers to the dilation of the pupils which is known to occur when an individual is particularly concentrating on a task. These factors can be integrated into a model to determine where an individual is focusing their attention on the screen. While some companies and organizations have attempted to create models similar to the one being described, they are generally suited for business applications and thus are meant to apply to the general population, not ADHD students. Most business applications (and applications monitoring students like Proctorio or Lockdown Browser) fail to account for the wide variation in human behavior and how different individuals look and engage with their virtual environment, thus flagging certain types of innocent behavior (such as looking away from the screen for a split second) as suspicious behavior at best or (in the case of students) cheating in the worst case. Moreover, applications for monitoring students (Proctorio or Lockdown Browser) are built with the purpose of monitoring students for standardized tests, not for virtual environments where students engage with instructors to learn new material, and definitely not for students who have ADHD. Combined with audio processing and analysis, this data can assist with structuring a learning simulation with minimal background noise and distraction, in order to ensure an optimized learning experience and performance for students with ADHD.

UI/UX

Motivation is essential for students with ADHD to succeed in an e-learning environment. Specifically, the inward social and biological drive is needed to endure distance learning [24].

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Simple and engaging user interfaces are necessary to keep students motivated by fostering the three components of motivation: autonomy, competence, and relatedness [32]. Autonomy refers to how people prefer to make their own choices, competence describes the feeling when people use their skills to succeed, and relatedness refers to how people are motivated through relationships with their peers [38]. These three important components originate from the Self-Determination Theory (SDT), a theory that explains humans' innate inclination to be productive and efficient [38]. Studies of e-learning show that these internal drives also outweigh external factors in a virtual classroom [12]. Therefore, it is important to integrate tools to socialize and to receive immediate feedback into an e-learning platform. Abilities such as liking comments or replying to a peer's comment can help students socialize in a productive manner while creating a virtual community [13]. Such components can also help close the gap between the teacher and the students, which can help create a safe environment for students [12]. On the other hand, reward systems and affirmative feedback can help students feel motivated to complete their lessons and coursework. Language apps such as Duolingo and Memrise implement a point system modularized into specific topics that help users visually see their progress as they learn, and benchmarks help users feel motivated to complete the individual "levels" [22]. The bright colors, game-like avatars, and point systems result in the gamification of education, which helps users stay engaged while learning a new language [22]. Implementing these engaging components into the user interface will greatly complement the attention level algorithm.

C. Ethical Considerations

An ethical issue with audio detection is the range of information the computer can gather from a voice sample. According to scientific studies, a single voice sample can reveal the user's ethnicity, background, age, and more [30]. In order to protect the user, it is necessary that the software either encrypts the audio or only analyzes audio on certain cues [30]. Additionally, users must be notified before e-learning sessions that they would be providing video and audio data to the software in order to maintain transparency and trust between the software and the user [7].

D. Research Significance

Countless advancements have been made in both education and technology to improve students' overall learning experiences. However, when analyzed further, there is a notable gap between the technology available and educational quality for students with ADHD. This research has the potential to fill the gap between these two disciplines, using current and creating new technology, specifically through machine learning, to improve the standard of online education specifically for students with ADHD. Through this project, instructors can gauge students' attention span and engagement during class, using this to modify the curriculum to provide the best experience for the student. Many symptoms of ADHD such as impulsivity, hyperactivity and boredom during lessons can be recognized and managed by the model, combining reinforcement learning with an engaging front end user experience to allow students with ADHD to get the most from their educational experience.

III. DISCUSSION OF THE SUGGESTED MODELS & IMPLICATIONS

Video Processing Model

One method for processing video is to use a combination of encoder-decoder **recurrent neural networks** (RNNs) and, in particular, **long short-term memory** (LSTMs) architecture. Recurrent neural networks are excellent for video processing due to their ability to act on previous data. That is, RNNs can learn using a feedback loop where the output in one step is fed as input in the next step. Thus, by separating the video of a student into individual frames, the RNN units can process the image of the student and, using the data from previous images, determine whether there is a change in the attention of the student (i.e, a change in eye movement, posture, head movement, etc).

While RNNs are suited to processing sequences of data, it is possible that the change between the frames of the video will vary greatly. Thus, using a long short-term memory architecture will ensure that RNN will not unnecessarily backpropagate changes that will have minimal effect or changes that were already observed. Another issue that must be considered is that certain aspects of the image should be focused on more than others. Thus, by using an **attention model** (a machine learning model that mimics how human brains distribute their attention spans when examining an image or hearing a sound), the RNN can weigh certain aspects of the image more, concentrating on aspects of the image that might be more indicative of the student's current attention level. This is especially useful on an individual level since even though ADHD students might be prone to certain types of behavior more than regular students, there is still wide variation in the behavior of individual ADHD students, and so any model which can adapt to the idiosyncrasies that result from ADHD in particular students will be more beneficial than a model which cannot understand student behavior on an individual level.

In terms of video processing, the model should produce an output indicating the student's "level of attention." This can be interpreted as a probability which can be combined with the audio processing model to determine the student's attention level. Through a user interface, an instructor can be notified by the model when the student's attention level is below a certain threshold. An instructor can indicate whether the algorithm correctly identified a student as paying attention or not. This instructor feedback can be utilized by the model to correct its predictions and, like the instructor, become familiar with certain cues that indicate whether the particular student is paying attention or not.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

Audio Processing Model

When developing the audio processing model, it is imperative to first consider the factors that will be involved when analyzing audio. Factors such as background noise are difficult to identify, and they are often inevitable when learning from home. Instead, the audio processing model will focus on the response of the student, analyzing its similarities with the teacher's question. This model follows the guidelines for designing software for students with ADHD by analyzing relatedness of responses and reaction time.

The audio processing algorithm will use a raw audio sample and convert it into a probability, from 0 to 1, that indicates the likelihood of a student being distracted/inattentive. Two factors can be used to detect whether a student is distracted: their response time and their answer's relevance to a question asked. These two factors require an algorithm that converts audio into speech. In order to determine the two factors, a Connectionist Temporal Classification (CTC) can be integrated into the model. A CTC takes in a short section of an audio sample, in the form of a spectrogram, and pushes it into a neural network that determines the likelihood of the audio sample being each of the 26 alphabets—or a space [2] This process is repeated, cutting the audio's spectrogram into short sections, until each section of the graph is processed by the neural network [2]. Due to the audio being cut into short snippets, the output will likely produce an output such as "hhhheeeeellllllooooo" instead of "hello" [2]. In order to solve this issue, another function will likely need to remove the adjacent duplicate characters and compare the output to a set of words in a dataset of the American language. Fortunately, there are a multitude of expansive datasets available on the internet, so it will be relatively simple to process the output of the CTC. The CTC itself can also determine the time at which a student begins to speak by marking the time point of the first character identified in the audio sample. This can be used to determine the response time of the student. Another potential problem the algorithm might face is deciding between two similar words. Some consonants are pronounced differently in different words, such as the 'c' in "cake" and "city." Also, the deletion of duplicate characters would be an issue when the model is deciding between two similarly structured words, such as differentiating between "look" and "lock." To determine the correct word, the model will likely need to use a natural language processing algorithm to pick the word that is more likely to be used in the context of the user's input.

The CTC and the neural network used with the CTC only produce phrases/sentences. In order to determine the relevance of a student's response, a natural language processing algorithm can be used. For example, the student's response and the teacher's questions can be compared by using the GloVe learning algorithm. The algorithm takes in a word and produces a vector for that specific word [25]. Similarities of words can then be compared by taking the distance between two vectors: a small distance between two vectors indicates a related word, while a large distance between two vectors indicates opposite or unrelated words [25]. Using this learning algorithm, the audio processing model compares words from the student and teacher's responses; however, certain words might need to have a stronger "weight" when determining when two sentences are related overall. If the sum of similarities of words is used as an indication of relatedness, common words in sentences, such as "a", "the", and "and" might skew the similarity index of two sentences. In order to combat this, the model will place a weight, or increased importance, on specific words related to the question the teacher asks in the form of a constant multiplied to the numerical representation of important words in the sentence. On the other hand, the model will also need to either remove frequent words or decrease its weightage when calculating the overall similarity of two phrases/sentences.

Architecture

The audio model will consist of an algorithm that analyzes the relatedness between the teacher's question or prompt and the user's response, as described in the audio section. Additionally, the user's camera input (video) will be analyzed using recurrent neural networks and long short-term memory models. These two audio and video algorithms will individually produce a set of values from 0 to 1 indicating the distraction level of the student, where each value corresponds with a factor, such as eye movement, input relatedness, and rapid movements. Each of these values will be averaged to produce a final value between 0-1 indicating the likelihood or extent to which the algorithm believes the student is unfocused. However, this average will weigh each factor, producing a weighted probability, as the factors likely do not have equal weights in a real virtual classroom setting. In order to weigh each of these factors, a genetic algorithm will be used. This algorithm will produce coefficients corresponding to each factor's value, or the weight of each factor.

A genetic algorithm mimics the process of natural selection by breeding generations of children, where each child for this model represents a set of coefficients corresponding to the factors produced by the video and audio models. Each child breeds to create a new generation that consists of the offspring of the old generation. Pairs of coefficients then "breed" again by producing a new set of coefficients that are produced by selecting some of the coefficients from one parent and selecting some of the coefficients of the other. However, the "fitter," or the more accurate, set of coefficients are more likely to breed and pass on their characteristics to the next generation, so overall, each generation improves the set of coefficients. For this algorithm, the "better" set of coefficients will be the set that is more accurate in predicting the distraction level of the student. Because the model trains on unlabeled data and uses reinforced learning, testing the effectiveness, or fitness, of each set of coefficients depends on how often the teacher chooses to verify the model's outputs,

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

so the model would need a maximum of a week of classes to find the optimal parameters with an accuracy above 75 percent. However, it is possible to save the model throughout multiple days, so the model would learn how to better combine the different probabilities produced by the algorithms as the teacher and students progress through the course.

UI/UX

Though a detailed implementation of a user interface and user experience will vary in virtual educational environments for ADHD students, it will be sufficient to examine the necessary components for an engaging user interface based on previous studies. No one user interface will be particularly suitable for all ADHD students, but certain recommendations can be made based on research on how ADHD individuals interact with their computer screens.

Firstly, any user interface that seeks to benefit ADHD students should seek to engage its users. ADHD students will be more interested in interacting with a "gamified" user interface that will reward them for engaging with it [40]. Following good principles of design, any user interface should also attempt to be simple to navigate. While this is easier said than done, this principle is far more important when designing user interfaces for ADHD students. If an interface is confusing, cluttered, or demands a substantial amount of cognitive resources to use and navigate, then it might stifle the internal motivation of ADHD students to engage with it. Finally, a user interface for ADHD students should seek to require few prerequisites for its usage. Since ADHD students might be experiencing a substantial amount of stimulus from their physical environment (background noise, staring out the window, etc), any interface that is designed for ADHD students should try to avoid requiring extraneous peripherals such as headphones or more than one device (i.e, a student should just use their computer or phone, not both). This minimizes distractions for the students since it requires less cognitive resources [20].

IV. RECOMMENDATION

To implement this algorithm, two separate models would need to first be developed: the video analysis model, and the audio processing model. The video of the student participating in virtual learning would be the input for the video analysis model, and the model would use a combination of an RNN and LSTM to output a value that represents the probability of a student being focused. On the other hand, the audio from the student would be the input for the audio processing model that first uses a CTC and neural network to convert pieces of audio into text and then uses the GloVe learning algorithm to determine the relevance of the student's response to the teacher's verbal cues. The audio processing model will then output a probability of the student being focused, which is directly proportional to the relevance determined by the model. The genetic algorithm then should use the two probabilities from the audio and video processing model to generate a pair of coefficients. These coefficients correspond to the weights of the audio and video processing models, and they would be used to compute a final weighted average representing the overall probability that the student is focused. This probability would then be sent to the frontend of the application in which it is implemented, alerting both the student and teacher of this probability. To improve the frontend of this application, the e-learning platform should include a reward system to maintain the motivation of the student while also ensuring that the design of the platform is clean and simple to avoid possible distractors.

The reinforcement learning algorithm detailed in this paper is recommended as the most effective model, which, when combined with an engaging user interface, can improve the overall educational experience of students with ADHD. The model uses continuously updated audio and video data to analyze the distraction level of a student with ADHD, which can help an instructor keep their student on task; the user interface is customized to appeal to students with ADHD to increase their attention span which subsequently allows them to retain more information. Quantitatively, this model separately gauges attention and distractions through the video and audio analysis, and uses weighted probabilities to calculate the final attention span level of a student based on the data. Qualitatively, this can be used to enhance the UI, curriculum and lesson to engage the student and help them better understand the information over an online setting.

Although reinforcement learning was used as the most effective category of algorithms for this project, it is important to note that the model does require a lot of computation power and data; as the amount of data increases, its accuracy and performance will increase as well. This can cause delayed results, which is unfavorable in an educational environment as an educator needs to know the immediate distraction level of their student at any given time. However, in some cases the model may lead to an overload of output states, which may cause skewed or unclear results [15]. Therefore, it is recommended to try this model with different types of algorithms to evaluate performance and accuracy of the model in different virtual environments, keeping in mind advantages and drawbacks of supervised and unsupervised learning as detailed in the literature review. This research mainly benefits students with ADHD or with learning disorders; however it can be used in any online setting to gauge distraction level. It should be noted that the aspects used to consider attention span are based on symptoms of ADHD, so the model may not perform as highly for students with other learning disorders, or disorders that present a separate set of symptoms.

V. CONCLUSION

Virtual education for students with ADHD is clearly challenging. Not only are reliable algorithms required to ensure the student is paying attention to the content and the instructor, but these algorithms must adhere to certain ethical standards while providing both the instructor and the student with a streamlined learning experience.

Retrieval Number: 100.1/ijrte.A59650510121
DOI: 10.35940/ijrte.A5965.0510121

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication

It is imperative in an Internet-connected world and in this pandemic era that virtual environments are inclusive of all types of users and that these environments can be effective mediums for education. In this paper, a literature review was conducted to address the advances made in virtual learning models for students. This literature review detailed ethical issues, future research opportunities for online education, and new insights the paper will bring to AI-powered virtual environments for education. Additionally, a machine learning model using an RNN and LSTM to process audio and visual data of ADHD students was suggested. Furthermore, the paper presented an outline for integrating this model into a UI/UX that would engage ADHD students while providing instructors with critical information on the student's engagement levels, and what such a UI/UX should aspire to accomplish. While this paper has addressed the steps that should be taken to provide a streamlined learning experience for virtual ADHD students, the software models provided should be implemented and tested in an actual virtual environment. Furthermore, future studies should seek to understand how virtual user interfaces and their design affect ADHD students' engagement and learning. While our recommendation outlines some guidelines for designing user interfaces for non-neurotypical students, these guidelines were broad and generic. Additionally, the machine learning models proposed require a diverse and large corpus of data in order to produce accurate interpretations of audio and visual data. Thus, studies on ADHD students and virtual learning environments should investigate a variety of machine-learning model that can operate with a smaller and possibly less diverse dataset than the one proposed in this paper. In essence, this paper demonstrates that a theoretical virtual learning environment for ADHD students is possible to create with the help of artificial intelligence and that future research should aim to investigate these models in practice and evaluate their implications for the education of ADHD students in virtual settings.

APPENDIX

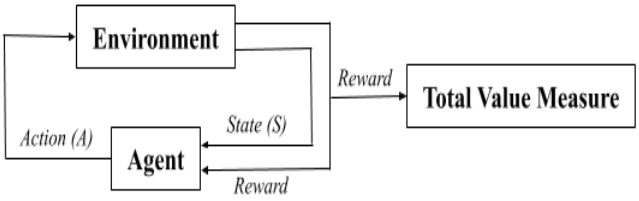

Figure 1: Basic Reinforcement Learning Model

Figure 1 illustrates a reinforcement model with the environment, agent, and total value. Arrows indicate the flow of the model. It initializes with an environment and agent, where the agent chooses an action to implement in the environment, through which a state and reward is generated; the state goes back to the agent to determine the next action, and the reward is added to the total value measure. This figure was created on Google Drawings, by the writers of this paper.

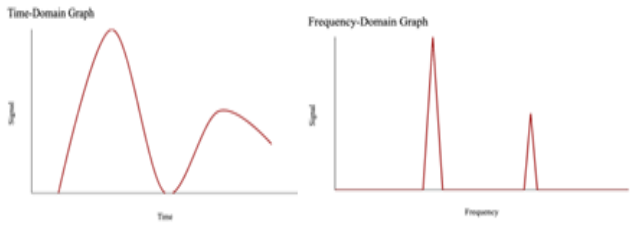

Figure 2: Visual comparison between time-domain graphs and frequency-domain graphs

Figure 2 is a comparison of the 2 types of data processing domain graphs: the time-domain and frequency-domain graph. Both graphs are in terms of the signal, although the time-domain graph evaluates the signal over time and the frequency-domain graph evaluates the signal for every frequency. The comparison presents the difference in shape, and shows a generic graph for each domain. This figure was created on Google Sheets, by the writers of this paper.

REFERENCES

1. P., By, & Packt. (2018, April 4). *Convolutional Neural Networks with Reinforcement Learning*. Packt Hub. https://hub.packtpub.com/convolutional-neural-networks-reinforcement-learning/.
2. Ageitgey. (2016, December 23). *Machine Learning is Fun Part 6: How to do Speech Recognition with Deep Learning*. Medium. https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a .
3. Angel, T. (2020, September 5). *Everything You Need to Know About ADHD*. Healthline. https://www.healthline.com/health/adhd.
4. Aristizabal, A. (2020, October 19). *Understanding Reinforcement Learning Hands-on: Non-Stationarity*. Medium. https://towardsdatascience.com/understanding-reinforcement-learning-hands-on-part-3-non-stationarity-544ed094b55.
5. Kali. (2020, December 17). *The Future Of Education And Technology*. eLearning Industry. https://elearningindustry.com/future-of-education-and-technology.
6. Centers for Disease Control and Prevention. (2021, January 26). *School changes - helping children with ADHD*. Centers for Disease Control and Prevention. https://www.cdc.gov/ncbddd/adhd/features/adhd-and-school-changes.html.
7. *Convolutional Neural Networks for Eye Tracking Algorithm.* (n.d.). http://stanford.edu/class/ee267/Spring2018/report_griffin_ramirez.pdf.
8. Goodfellow et al. (2016). *Deep Learning.* MIT Press. http://www.deeplearningbook.org/
9. Dezfouli, A., & Balleine, B. W. (2012, April). *Habits, action sequences and reinforcement learning*. The European journal of neuroscience. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3325518/.
10. Soni, Devin. (2020, July 21). *Supervised vs. Unsupervised Learning*. Medium. https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d.
11. Voskoglou, Christina. (2019, December 11). *What is the best programming language for Machine Learning?* Medium. https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7.
12. Fandiño, F. G. E., & Velandia, A. J. S. (2020, August 6). *How an online tutor motivates E-learning English*. Heliyon. https://www.sciencedirect.com/science/article/pii/S2405844020314742.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication

13. Ferlazzo, L. (2021, March 5). *Four Ways to Help Students Feel Intrinsically Motivated to Do Distance Learning (Opinion)*. Education Week. https://www.edweek.org/teaching-learning/opinion-four-ways-to-help-students-feel-intrinsically-motivated-to-do-distance-learning/2020/04.
14. Hannah, J. (2019, October 2). *What Is A User Interface, And What Are The Elements That Comprise one?* https://careerfoundry.com/en/blog/ui-design/what-is-a-user-interface/.
15. Joy, A. (2020, June 11). *Pros And Cons Of Reinforcement Learning*. Pythonista Planet. https://pythonistaplanet.com/pros-and-cons-of-reinforcement-learning/.
16. Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996, May). *Reinforcement Learning A Survey*. https://www.cs.cmu.edu/~tom/10701_sp11/slides/Kaelbling.pdf.
17. Krishna. (n.d.). *Supervised Machine Learning: What is, Algorithms, Example*. https://www.guru99.com/supervised-machine-learning.html.
18. Krishna. (n.d.). *Unsupervised Machine Learning: What is, Algorithms, Example*. https://www.guru99.com/unsupervised-machine-learning.html.
19. McCue, C., Cates, J., & Whitaker, R. (2017). Modeling Algorithm. ScienceDirect. https://www.sciencedirect.com/topics/computer-science/modeling-algorithm.
20. McKnight, L. (n.d.). *Designing for ADHD: in search of guidelines*. UIowa. http://homepage.divms.uiowa.edu/~hourcade/idc2010-myw/mcknight.pdf.
21. Mendels, G. (2019, November 18). *How to apply machine learning and deep learning methods to audio analysis*. Medium. https://towardsdatascience.com/how-to-apply-machine-learning-and-deep-learning-methods-to-audio-analysis-615e286fcbbc.
22. Mira, T. (2020, January 5). *7 UX & UI trends on e-Learning*. Medium. https://medium.com/dsgnrs/7-ux-ui-trends-on-elearning-bdd623b7baaa.
23. Singh, Jyotika, (Correspondent), D. W., & (Correspondent), C. H. (n.d.). *An introduction to audio processing and machine learning using Python*. Opensource.com. https://opensource.com/article/19/9/audio-processing-machine-learning-python.
24. Ovesleová, H. (2015, July 21). *E-Learning Platforms and Lacking Motivation in Students: Concept of Adaptable UI for Online Courses.* Springer Link. https://link.springer.com/chapter/10.1007/978-3-319-20889-3_21#Sec2.
25. Pennington, J. (n.d.). *GloVe: Global Vectors for Word Representation.* https://nlp.stanford.edu/projects/glove/.
26. Ray, S. (2020, December 23). *Commonly Used Machine Learning Algorithms: Data Science*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/.
27. Reckdahl, K. (2020, November 12). *5 Ways to Support Kids With ADHD During Remote Learning*. Edutopia. https://www.edutopia.org/article/5-ways-support-kids-adhd-during-remote-learning.
28. Morkos, Ragi. *Sampling Rates, Sample Depths, and Bit Rates: Basic Audio Concepts*. Voci. (n.d.). https://www.vocitec.com/docs-tools/blog/sampling-rates-sample-depths-and-bit-rates-basic-audio-concepts.
29. Shaik, F. (2020, December 23). *Audio Data: Audio/Voice Data analysis Using Deep Learning*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2017/08/audio-voice-processing-deep-learning/.
30. Shojaeizadeh, M., Djamasbi, S., Paffenroth, R. C., & Trapp, A. C. (2018, October 31). *Detecting task demand via an eye tracking machine learning system*. Decision Support Systems. https://www.sciencedirect.com/science/article/abs/pii/S0167923618301696.
31. *Time Domain Analysis vs Frequency Domain Analysis: A Guide and Comparison*. Time Domain Analysis vs Frequency Domain Analysis: A Guide and Comparison | Advanced PCB Design Blog. (2021, January 25). https://resources.pcb.cadence.com/blog/2020-time-domain-analysis-vs-frequency-domain-analysis-a-guide-and-comparison.
32. *Machine Learning Algorithms: What is a Neural Network?* (2018, November 19). https://www.verypossible.com/insights/machine-learning-algorithms-what-is-a-neural-network.
33. Smoot, Jeff. *Understanding Audio Frequency Range in Audio Design*. CUI Devices. (2020, June 4). https://www.cuidevices.com/blog/understanding-audio-frequency-range-in-audio-design.
34. Reynolds, Jennifer. U.S. News & World Report. (n.d.). *Can Technology Help Kids With ADHD Stay Focused?* U.S. News & World Report. https://health.usnews.com/health-care/patient-advice/articles/2017-05-03/can-technology-help-kids-with-adhd-stay-focused.
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (n.d.). *Attention Is All You Need.* Google. https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
36. Victoria, E. (2020, November 12). *Online learning vs. in-person classes – what's better?* EF. https://www.ef.edu/blog/language/online-learning-vs-in-person-classes-whats-better/.
37. *What is Machine Learning? A definition - Expert System*. Expert.ai. (2021, February 11). https://www.expert.ai/blog/machine-learning-definition/.
38. Wroten, C. (2018, April 10). *Motivate Your Learners! The Self-Determination Theory for e-Learning*. eLearning Industry. https://elearningindustry.com/motivate-learners-self-determination-theory-e-learning.
39. Granet, D. B., Gomi, C. F., Ventura, R., & Miller-Scholte, A. (2005). The relationship between convergence insufficiency and ADHD. *Strabismus*, *13*(4), 163–168. https://doi.org/10.1080/09273970500455436
40. Kusumasari, Dian & Junaedi, Danang & Kaburuan, Emil. (2018). Designing an interactive learning application for ADHD children. MATEC Web of Conferences. 197. 16008. 10.1051/matecconf/201819716008.

AUTHORS PROFILE

Srivi Balaji, is a 11th grader at Meridian World School in Round Rock, Texas. She has been independently coding for 4 years and wishes to pursue computer science and software development. She collaborated with a professor at Austin Community College to evaluate self-care and mental health practices of helpers-in-training, specifically during the social distancing period. By using her experience in technology and data analysis she was able to connect different disciplines to improve our understanding and use of personal care. Srivi currently works at Coding with Kids as a Coding Instructor, teaching elementary and middle school students to code in Python and Scratch. Her work with students and technology initiated her interest in this research field. In her spare time, Srivi loves music, research and Model United Nations.

Meghana Gopannagari, is an 11th grader attending Thomas Jefferson High School for Science and Technology. She is interested in computer science and assistive technology and hopes to pursue the intersection of these fields in high school and beyond. Previously, she designed a reading tool assistive technology for students with dyslexia by interviewing and closely working with her tutee. She continues to explore her interest in assistive technology and computer science by actively participating in the assistive technology club where she works with other members of the club to create enriching websites for students in her county. In her free time, she enjoys tutoring, running and participating in hackathons.

Svanik Sharma, is a junior attending Vandegrift High School in Austin, Texas. Svanik is an aspiring computer scientist who is looking to apply his skills to neurology. Svanik has participated in UIL Computer Science and Mathematics in his time at high school and has also been part of Mu Alpha Theta. Currently, Svanik participates in FTC Robotics as part of the software team for SnakeByte 4546. Svanik previously had an internship for the City of Austin to design software to benefit small businesses during the pandemic. In his spare time, he likes to read non-fiction and plays guitar and piano.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication

Preethi Rajgopal is a rising senior at the Kelley School of Business at Indiana University. She has previously worked with startups in the education technology space and has served as an educator in various contexts. She more recently gave a talk at SXSW EDU about how to improve entrepreneurship education for youth. She currently helps instruct a class about social connection for freshman and transfer students at IU. She also previously served as an educator at an MIT summer program as well as a program facilitator for an education non-profit called 3 Day Startup. In her free time, she likes to draw, read poetry, and cook.