Modeling Students' Attention in the Classroom using Eyetrackers

Narayanan Veliyath
Computer Science
Georgia Southern University
nv00321@georgiasouthern.edu

Pradipta De
Computer Science
Georgia Southern University
pde@georgiasouthern.edu

Andrew A. Allen
Computer Science
Georgia Southern University
andrewallen@georgiasouthern.edu

Charles B. Hodges
College of Education
Georgia Southern University
chodges@georgiasouthern.edu

Aniruddha Mitra
Mechanical Engineering
Georgia Southern University
mitra@georgiasouthern.edu

ABSTRACT

The process of learning is not merely determined by what the instructor teaches, but also by how the student receives that information. An attentive student will naturally be more open to obtaining knowledge than a bored or frustrated student. In recent years, tools such as skin temperature measurements and body posture calculations have been developed for the purpose of determining a student's affect, or emotional state of mind. However, measuring eye-gaze data is particularly noteworthy in that it can collect measurements non-intrusively, while also being relatively simple to set up and use. This paper details how data obtained from an eye-tracker can indeed be used to predict a student's attention as a measure of affect over the course of a class. From this research, an accuracy of 77% was achieved using the Extreme Gradient Boosting technique of machine learning. The outcome indicates that eye-gaze can be indeed used as a basis for constructing a predictive model.

CCS CONCEPTS

• **Applied Computing** → **Education**; • **Computing Methodologies** → *Machine Learning*;

KEYWORDS

Affective Computing, Attention, Eyetracking, Machine Learning

ACM Reference Format:
Narayanan Veliyath, Pradipta De, Andrew A. Allen, Charles B. Hodges, and Aniruddha Mitra. 2019. Modeling Students' Attention in the Classroom using Eyetrackers. In *2019 ACM Southeast Conference (ACMSE 2019), April 18–20, 2019, Kennesaw, GA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3299815.3314424

1 INTRODUCTION

Affect is a term with origins in psychology, referring to a being's emotions, motivations, and interests. Affective learning is learning that considers these points in addition to the material. While normal education tends to focus on merely *what* is being taught, affective learning additionally chooses to tackle the issue of *how is the student receiving it?* [8]. This can be considered especially vital because interest is a vital part of learning. An engaged student will not only be more open to gaining knowledge, but is much more likely to retain that information as well. However, this task can be especially challenging because gauging emotions is not a straightforward task. There is no single guide for determining a person's emotional state of mind.

As there is no standard method for estimating attention, many approaches have been developed to solve the problem. The methods employed range vastly in both scope and complexity. Some methods involve a through overview of subjects, taking in measurements such as skin conductive response, body temperature and brainwave readings. While these measurements are thorough and extensive, they are also intrusive. Intrusive measures can often be uncomfortable for the user, which in turn can lead to errors in the collected data. Additionally, they often require more physical setup, adding an additional cost to be considered. As a result, other researchers have turned toward less-intrusive approaches instead. These methods of collection require less active thought from the user, and can still generate valuable results. Body posture analysis, facial analysis, and gaze collection are example of such methods.

The research conducted here focuses on gaze data obtained through an eyetracker. Gaze data refers to the points in space where a user's vision is focused. Gaze data is significant in that it can yield accurate predictions about a user's attention. This method is also based around the notion of being almost completely non-intrusive. While there are other means of obtaining gaze data, some of them fall into the 'intrusive collection' category [12]. Furthermore, there is relatively little effort required in setting up such an eyetracker, allowing for relative ease of use. The goal of this research is to create a predictive model of students' attention in a classroom setting.

The eye-tracker used to collect the eye-gaze data is known as a Tobii Eyetracker, model 4c. An example of this type of eyetracker can be seen in 1. The eyetracker is attached to the base of the computer monitor, and is connected into the computer via USB.

ACMSE 2019, April 18–20, 2019, Kennesaw, GA, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6251-1/19/04…$15.00
https://doi.org/10.1145/3299815.3314424

Additionally, software was developed to collect certain information from the users' computers. This additional information collected would act as the features needed for training a model. As part of the collection software, a GUI was created to get self-reported scores from the users. This GUI appeared in the form of a Likert Scale popup, which the users would click on to respond to. The final step of the approach involved using machine learning to generate models, as well as comparing generated models against each other.

Figure 1: Tobii 4c Eyetracker

The results generated by the models show that it is indeed possible to make reliable predictions of a student's attention. One model was able to reach an accuracy of 77%. This not only proves that gaze data can be a worthwhile source of affect, but also that models generated from non-intrusive methods have the potential to be on par with intrusive techniques. Perhaps most importantly, these results are in a form that can be easily returned to an instructor, allowing them to makes decisions and adjustments based on the ever-changing affect of a classroom.

The rest of the paper is organized as follows. In Section 2, we present the literature related to measuring and use of affective states of students during learning. In Section 3, we present the details of our methodology in collecting and analyzing the data gathered in a classroom. Section 4 presents the data collected from the eyetrackers and classrooms, and introduces the machine learning models to be used. Section 5 reveals the results of the machine learning models through multiple metrics. It also goes over the feature importance, and presents another model that considers the use of 'No Response' labels. In Section 6 the challenges and limitations encountered during this experiment are discussed. Finally, Section 7 concludes the paper and addresses future works.

2 RELATED WORKS

While affect is certainly a key factor in the learning process, it has not always been openly considered as such. Even in the 2000s, the effects of affect on attention was a topic only beginning to be examined. In 2007, Smallwood, Fishman, and Schooler not only showed the detrimental effects of mind wandering, but also how it could cascade into further problems [14]. Their results indicated that failure to catch dwindling attention could cause a negative feedback spiral of continually diminishing attention. This in turn results in a student being unable to return to an attentive state by themselves.

2.1 Intelligent Tutoring Systems using Affective States

Lehman et al. went on to study the impact of affect during 1-on-1 scenarios with a human tutor, as well as with an Intelligent Tutoring System (ITS) [9]. While their experiment did not directly consider a classroom setting, the outcome did confirm that there were marked differences between the two scenarios. When acting with an expert human tutors, subjects felt more 'personal' attention, and were much less likely to lose focus. However, while working with an ITS, negative affect was much more prominent, in chief due to the lack of personal feeling. This second part is much more vital to consider in large classrooms, as it is simply not possible for an educator to give attention to all students individually.

ITSs have become an increasingly important tool for the goal of predicting affect. An ITS is a piece of software that aims to provide instruction and feedback to students. Much like a physical tutor would, an ITS dispenses knowledge to its users. However, unlike online courses or textbooks, many intelligent tutoring systems are also designed with the idea of being able to gauge its student's affect. Just like a human tutor may notice that a student would be losing interest, ITS have become increasingly equipped to not only detect such students but to also return them to an attentive state as well.

One of the more fundamental works in the ITS field came in the early 2000s. Known as AutoTutor, this ITS became well known for its use of natural language to converse with its user [6]. By speaking directly to the student, AutoTutor aimed to keep engagement levels high in order to maximize learning. Rather than reacting to a lack of attention, this ITS instead continuously engaged the user, chiefly in the form of questions. This method proved to be effective, as other studies on the same ITS revealed a strong correlation between attention level and learning [4].

Another well-known ITS, MetaTutor, has also been used as the basis for many experiments [1]. While AutoTutor attempts to directly engage the student, MetaTutor instead encourages its users to regulate their own learning. It then adapts to this self-learning in an effort to better assist the student. As a result, MetaTutor is more straightforward in how it operates compared to AutoTutor. However, this also means that it can be considered a more natural and normal environment for testing.

Jaques et al. used MetaTutor to predict affect, more specifically the emotions of 'curiosity' and 'boredom' [7]. Additionally, they noted that other prior studies tended to use physical features such as pupil dilation, which is not necessarily correlated with eye-gaze data. This in turn lead them to explore the effectiveness of gaze data alone as a predictor. While they were successful in proving their hypothesis, it should be noted that the features extracted were dependent on the MetaTutor environment. This does indeed demonstrate the usefulness of the approach, but also limits the scope of use, as it may not pertain to other similar areas.

While some researchers use the more popular and readily available intelligent tutoring systems, others chose instead to develop their own systems instead. D'Mello et al. was one such team, creating their own ITS to work with. Aptly named GazeTutor, the ITS uses eye tracking software to monitor a student's gaze and predict their affect [5]. When the student is determined to be bored or disinterested, GazeTutor will then attempt to directly reengage them, in the hopes of reigniting their interest. D'Mello was able to show that such prompts were able to successfully return a student to a state of attention without causing them to become irritated at the system. However, he did also note that while such prompts did solve the problem of attention, they were also unable to remedy the underlying motivational issues. In a class setting, this task would fall to the teacher. If the student could be accurately identified as ‘not interested’, the teacher could take actions to try and motivate them.

Exploratory Learning Environments (ELE) are an offshoot of ITSs. While a standard ITS has a structured and set system, ELEs tend to be more free, allowing for more open exploration of the subject. While this does offer its own advantages, it also adds a level of difficulty in capturing attention patterns. Conati and Merten used gaze information with such an ELE, to show that gaze data could be used in coordination with time evidence [3]. This combination produced better results than either feature could alone. In addition, they tested for correlation between pupil dilation and cognitive action, and instead found no positive correlation between the two at the time. This adds further credence to the notion that gaze data can not only be an accurate predictor of affect, but also has the potential to exceed other methods.

2.2 Alternate Methods of Predicting Affect

While intelligent tutoring systems have gained a lot of attention for their use in predicting affect, they are far from the only means of doing so. Muldner et al. developed their own learning environment for the purposes of gauging positive and negative affective states through the use of an eye-tracker [11]. Their results did show that pupil dilation could indeed be used as a reliable feature for model generation. However, their work focused more on pupil response rather than gaze data. This does not negate the significance of their work, as it gave another proof that non-invasive methods of collecting data are becoming more reliable.

Tools have even developed to detect loss of attention while watching online videos [13]. Created by Sharma, Alavi, and Jermann, this tool was designed to draw the user’s attention back to the relevant portions of the video. A highlight would appear around the important area when the user’s attention was deemed to be below an average value, compared to other students watching the same video. Perhaps most importantly, this tool was shown to not only improve the user’s immediate attention, but also raised their average attention over a longer period as well.

D’Mello implemented software to detect mind wandering while reading text on a computer, through the use of gaze data. While the setup does require an extensive amount of preprocessing, the results it generated proved again the viability of gaze data as a predictor [2]. Even hardware initially designed for other purposes have been used repurposed for the goal at hand. The Kinect motion sensor was an attachment for the Xbox line of video game consoles, used for playing motion-based video games. In the hands of Zaletelj and KoÅąir however, it became a tool to predict affect from both body and facial features [15]. The Kinect was able to detect body posture, head angle, and even gaze information from multiple individuals at once. Just as importantly, it was able to generate reliable predictive results. However, analysis of the information received required extensive pre-processing to be used for machine learning. This adds another depth of difficulty and complexity, which can hinder its use for general purposes.

3 METHODOLOGY

The relevant details of the participants will be introduced. The physical setup of the experiment as well as the software involved will also be clarified. Lastly, the actions undertaken by the volunteers during the experiment will be explained. Figure 2 shows the experimental design for this research.

3.1 Participants

The participants were volunteers from undergraduate classes in the field of mechanical engineering. The class was composed of Junior level students, and had approximately 25 students in attendance. Each class lasted for approximately 90 minutes and met once a week. Of the 25 students, ten individuals volunteered during the semester. Six of the volunteers were male, and four were female.

All of the volunteers were told that an eyetracker would be placed in front of their computer monitor. This eyetracker would be used to collect their gaze data. Additionally, software would be installed on their machines to collect other information as well. They were also informed that every five minutes, a popup would appear on the screen. The volunteers were instructed to respond to the popup based on the question asked at the beginning of the study.

3.2 Physical Setup

The classroom was set up in the style of a computer lab. This setup is notably different from a standard classroom in that all students are provided with desktop computers at their desks. In this environment, a professor stands at the front of the classroom, and lectures to the students, often with the aid of PowerPoints or a whiteboard. Meanwhile, students sit at their desks with computers in front of them. They are able to use the computers freely, and also had the option of using physical notebooks and other tools as needed. This setup allows students to either pay attention to a lecture by directly looking at the instructor, or by following allowing with provided material on their computer.

Eyetrackers were physically secured to the base of each computer monitor. Each eyetracker was carefully placed so as to not impede nor overly distract any users. As the eyetrackers were locked into place at the time, students had to move to computers with eye trackers installed on them. This was determined to not have any detrimental effects on the students attention or behavior. The instructor did not have any special instructions nor did they have to make any alterations to their planned teaching schedule. All efforts were made to ensure that the eyetrackers did not disrupt the normal classroom process. This is particularly vital, as minimizing intrusion was a key aspect of the research.

3.3 Data Collection Software

For this research, software was implemented to collect, store, and parse through the information collected. From the Tobii eyetrackers, data related to the location of gaze on the screen was saved to a text file on a private server. Each text file corresponded to a single student on a given class day. The local timestamp, foreground application name, foreground application coordinates, and ground truth, or self-reported scores, were also saved to the same text file.

To collect the ground truth from the students, a pop-up GUI was generated. Every five minutes, this pop-up would appear at the top left of the computer screen with a Likert Scale on it. The software additionally performed a check on the user as they logged into the

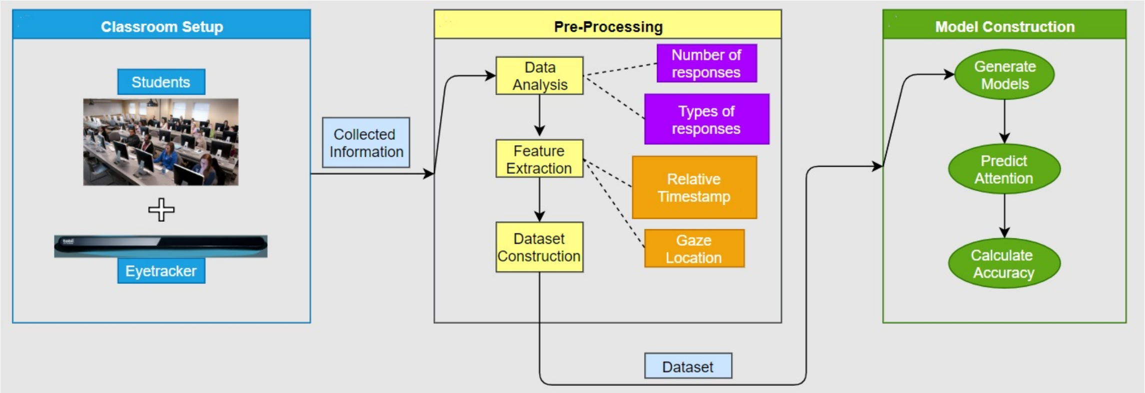

Figure 2: Data Process Diagram

computer. There were two conditions checked: If the user was in a given list of volunteers; Or if the time/date was appropriate for the correct class periods. If either of these conditions were not met, the software would immediately shut down, and no information would be collected. This was done to not only ensure security, but to also make sure that only the necessary data was collected. As the eye-tracker also shut down with the software, it also help prevent the setup from impacting students outside the experiment.

3.4 User Procedures

On the first day, volunteers calibrated the software to match their eyes. This was done to ensure that the eyetrackers would capture their eyes and eye movements as accurately as possible. Even with calibrations, there tended to be a small margin of error. However, the size of the error margin was not found to have any significant impact on the data collected. The students were told that every five minutes, a survey would pop up on their screen. They should ideally respond to the survey based on the question asked at the beginning of the experiment.

The question asked to all the volunteers was “How engaging were the previous 5 minutes of class?". This question was designed to allow the students to respond less personally, and thus more naturally. As seen in Figure 3, each survey was based on a Likert Scale going from ‘1-10’. A ‘0’ option was also included as a default, if the subject did not feel like responding during that instance. Beyond these instructions, the students had no limitations on their actions. It was in fact encouraged that they act as naturally as possible.

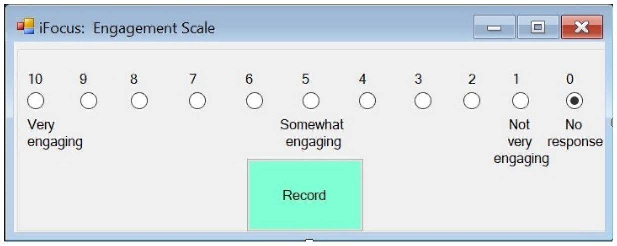

Figure 3: Self-report Popup

4 DATA PROCESSING

Five types of data were collected from the volunteers. From these data points, features for the machine learning models were extracted. The data was then analyzed and examined to identify any patterns. Finally, the techniques used for machine learning were introduced.

4.1 Data Collected from Students

As was mentioned before, five types of information were collected from each individual through the eyetracker, as well as from their computer. The first piece was the timestamp of when the ‘snapshot’ of the activity occurred. The Tobii Eyetracker 4c had a preset sampling frequency of 90Hz. However, due to limitations in writing the data, there was not necessarily 90 timestamps collected per second. Priority was given to time stamps that coincided with a user self-reported score, to allow the more important values to always be saved. The eye-gaze location on the screen was another data point that was recorded. The location was saved as an array of two points, with each point being the approximate pixel location of each eye on the screen. While there was some margin of error, proper calibration ensured that such problems would be greatly decreased. The foreground application name and screen coordinates were also gathered for analysis. The application name was used to confirm whether or nor the subject was looking at a website or application relevant to the course or area of study. The screen coordinates were used to verify that the user was indeed looking at the foreground application. Additionally, these coordinates would later become a feature of their own.

The final, and most vital, data point to be collected was each individual’s self-reported scores. Scores were broken into four general categories: ‘Direct responses’, which indicated any response with a number other than 0; ‘No response’, which refers to responses given as 0; ‘Missed response’, or a period where the user did not answer the survey at all; and ‘default responses’, referring to basic values written to the text file as a way to check the eyetracker was still running correctly. Of these four categories, only direct responses and ‘no responses’ were considered for feature use. It was determined that missed responses could not be accurately classified, as there would be no way to know if it indicated ‘high’ or ‘low’ user attention. Table 1 briefly outlines the collected information with a explanation for each data type.

4.2 Features Extracted

While five types of data values were obtained, four of them were used as features for this experiment: Eye-gaze location, foreground application being viewed, application coordinates, and the timestamp. Each of these features were pre-processed before machine learning. This is because the originally collected data were stored in continuous values too large or broad to be easily compared.

The eye-gaze coordinates were originally stored as pixel values based on the screen width and height. They were then grouped into one of four categories to indicate which quadrant of the screen the student was looking at, in order to determine if there was any correlation between gaze coordinates and attention. The first quadrant was located at the top left, and went counterclockwise, with the fourth and last quadrant situated at the top right.

The foreground application names were also processed to become binary values. Certain websites and applications were noted to be pertinent to school or educational matters. These were labeled as 'relevant' applications. All other applications and sites were labeled as 'irrelevant'. While there was an initial 'irrelevant' list, it was quickly discovered that the number of non-relevant applications greatly outpaced the number of relevant ones due to the open nature of the computer network. As such, it would be impossible to list every possible exception, creating the need to adjust the lists into a binary feature. However, there should still remain a strong link between low attention and 'not-relevant' applications.

The applications marked as 'relevant' were documented through manual input. This was achieved by having a separate program run through a list containing the unique websites and applications visited by each student, and tallying the number of occurrences per application. The resulting lists were then aggregated and then sorted by number of occurrences. The final list was reviewed by hand, and applications deemed to be most relevant to the classroom were inserted into the 'relevant' list.

The time stamp was written into the text file as UNIX time. UNIX time is the time elapsed in seconds, or milliseconds for this experiment, since 00:00 1 Jan, 1970. These UNIX values were then converted into local datetime values. The first value when the user first logged in and the software began collecting information was recorded. The time values where the pop-up appeared were also recorded, and the approximate minutes since the log in time was calculated. The difference in minutes was the actual value used for testing. This difference acts as a relative measure of time elapsed for each user. In a classroom, one would expect that the periods of high and low attentions would be similar for all students, or at least the majority of them.

The final feature to be considered was the area of the foreground application. The application coordinates received from each user's computer held the top left and bottom right coordinates. From these values, the area of the application being viewed was able to be calculated. It was believed that this held relevance as there could be a correlation between having multiple windows open and attention. With two windows being open side-by-side, for example, each window would have a small area than a single full size window.

4.3 Data Analysis

While the predicted values were initially in a multiclass format, they were processed to fit into a binary classification. This is because the basis of this research is to predict attention, and whether it is high or low. The exact level of attention, e.g. 5, 7, 10, has little relevance. Furthermore, a binary classifier would work better when returning the results to a professor or other interested individual. These results would be simpler to understand, without having to sacrifice any relevant details in the process. The number of responses in each category is shown in Table 2.

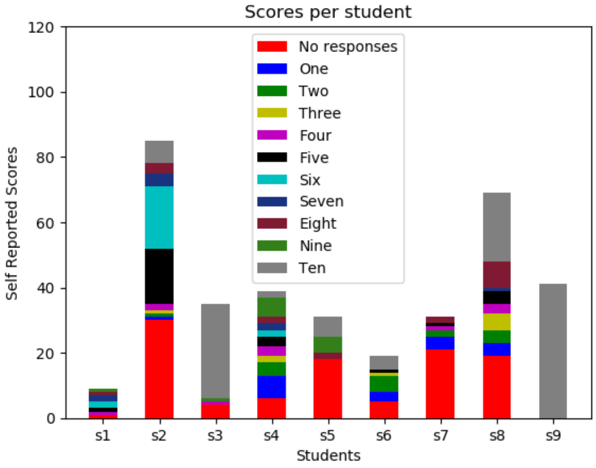

Figure 4: Self Reported Scores Per Student

Figure 4 shows the distribution of the scores per student before the class processing was conducted. As Figure 5 shows, there tends to be a respectable variety of responses per student. Almost all subjects provided a response that was able to divided into one of the three categories of (1-5), (6-1), and (0). The breakdown of these results is not perfectly balanced. However, this is to be expected from human subjects in an actual classroom environment.

Out of the ten volunteers, one subject had to be removed. The individual did not meet a satisfactory number of responses, and so would not prove beneficial to testing. All other subjects were kept, and used for the model generation by the machine learning. Subject 10 was also considered to removal as an outlier. This volunteer only responded with 'high attention' for the duration of the experiment. However, without outside knowledge of this individual, their responses must be taken without judgment. As a result, subject 10 remained as part of the data set.

4.4 Machine Learning Approach

For this research, an aggregate approach to the model was taken. This means that the data from all the students were combined, and then models were generated from the resulting dataset. Due to events outside of the experiment's control, not all subjects had equal amounts of responses. This would almost certainly lead to bias and errors in individualized testing. Additionally, in a college classroom setting, it was deemed unlikely that a personal approach would be more beneficial and accurate than an aggregate model.

Table 1: Collected Data and Descriptions

Collected Information	Description of information
Gaze location	Where on the screen the user was looking at
Timestamp	When the ‘snapshot’ was written measured in UNIX time
Foreground Application	What exactly was being viewed on the screen.
Application Coordinates	Where on the screen the application being viewed was located

Table 2: Classes and Number of Responses

Classifications	Number of Responses
No Response (0)	104
Low Attention (1-5)	82
High Attention (6-10)	173

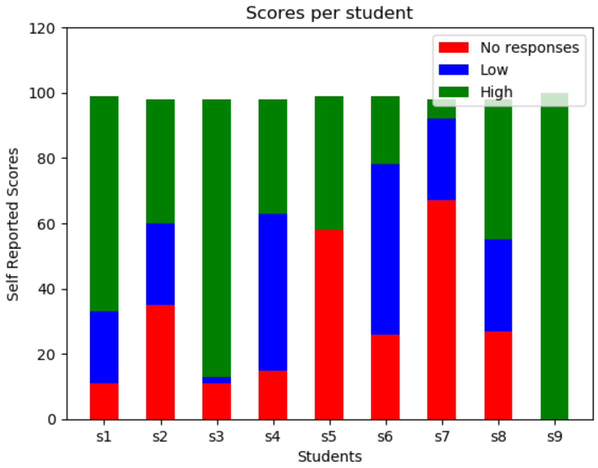

Figure 5: Self Reported Scores as Percentages Per Student

Four machine learning models were used for this research: Random Forest (RF), Support Vector Machines (SVM), Adaptive Boosting (AdaBoost), and Extreme Gradient Boosting (XGB). RFs are created by taking a multitude of decision trees, running them, and then taking the mean of their results. This is done primarily to combat overfitting, which decision trees can be prone to. While SVMs are a bit trickier to explain the nuances of, their operations can be simplified. At their core, an SVM simply tries to create a ‘line’ that separates all classes from each other neatly.

Adaboost and XGBoost both belong to the ensemble methods of machine learning. Ensemble methods are more advanced algorithms, used to improve on other model’s results. Adaboost and XGBoost belong to the bagging subset of ensemble methods, which have the effect of reducing innate bias in models. These techniques were chosen because they were simple enough to readily understood, while also having the ability to handle complex data. The use of neural networks was also considered. However, neural networks work best when they are fed a large amount of data. As that amount of data was unavailable at the time, neural networks were not chosen as a viable option.

5 RESULTS

The data collected was analyzed through four models. Each model was tested with the metrics of accuracy, precision, recall, and F-score. Once the models were generated, feature importance was calculated to determine the relative usefulness of each feature. Finally, an alternate theory was tested on a variation of the dataset.

5.1 Classification Metrics

Before delving into the machine learning itself, a brief overview of the metrics used will be provided. The first and most fundamental metric used was accuracy. This metric is the most straightforward, as it simply checks the number of correct predictions over the total number of predictions. The metrics of precision, recall, and F-score were also used as a further breakdown of the analysis. Precision is the ability of a model to identify true positives correctly. More exactly, precision checks the number of predictions that are actually positive, over all predicted positives. Recall, on the other hand, is used to determine how many true positives were found correctly. It can be calculated be taking the number of true positives over all actually positive values. F-score is a combination of precision and recall, and is used to get a quicker overview of both. These metrics were used as they also provide valuable insight into imbalanced datasets. They also offer an additional way to compare the results of various models.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

To ensure the most accurate results, K-fold cross validation was used. Cross validation is the process of splitting the data set into multiple subsets. One subset is used for testing, while the rest are instead used for training the model. This process is known as K-fold, because it is repeated K times. The repeated trials reduces the variability of the model’s accuracy, which can arise due to bias. The Leave-One-Out method of cross validation was also considered for this experiment. As it did not show any drastic changes compared to the K-fold cross validation, it was not used for testing. For these experiments, a K of 20 was chosen.

5.2 Performance Results

Table 3 shows the results of all the models using the metric of accuracy. A peak accuracy accuracy of .77 was achieved by Extreme Gradient Boosting. This result is considered fairly standard in the realm of machine learning. While not extraordinarily high, it did managed to outperform a base model by a significant amount. This base model only achieved an average accuracy of .52. This is a good indicator that a machine learning model is much more useful than a simple base model.

Table 3: Results for each Model

Aggregate	RF	SVM	Adaboost	XGB
Accuracy	.69	.73	.63	.77

Table 4 below shows the best precision, recall, and F score for all of the models. All model have similar precision scores. However, recall scores tended to have higher variance. This does lead XGB to a higher final F-score overall, being .01 higher than the next highest of .87. These numbers show support for the previous table, indicating why XGB and SVM tended to have the highest accuracy numbers.

Table 4: Precision, Recall and F-Score for each Model

Classifier	Precision	Recall	F-Score
Random Forest	.72	.73	.63
SVM	.78	.975	.87
Ada	.72	.58	.64
XGB	.78	1.0	.88

5.3 Feature Importance

Feature importance was also calculated as part of this research. Figure 6 shows a pie chart representing the relative importance of each feature. The feature importance was drawn from the XGB model. As the figure indicates, timestamps were the most important feature by a decent margin, holding a little under 50% of the importance. This is in line with previous assumptions, as in a classroom setting one would expect many students to have the roughly the same affect at the same times. On the other hand, relevant applications were not as useful as initially thought, being only 9 percent of the total importance. This is most likely due to extreme variance in what each person was looking at, allowing for little correlation between a self-reported score and what they were looking at during that time.

The gaze location was the second best feature, having an importance of .31. This illustrates that gaze features do hold a not-insignificant amount of importance, and do help greatly in predicting attention. While it might not be the best predictor on its own, it can help to supplement other features to produce a better model. The last feature was the application area which held an importance of 14%. While having more of an impact than the application type, it was not nearly as vital as the gaze location or timestamp.

Once Extreme Gradient Boosting was identified as the overall best classifier out of the four, parameter tweaking was done to attempt to improve its results. These parameters included the learning objective, learning rate, and base score. The parameter that yielded the most difference was the learning rate. Learning rate refers to the extent of which the weights in a loss gradient function are adjusted. It allows for tuning of the model without needing to adjust any classes or features. However, manipulating the learning rate is not an action without fail. Selecting a learning rate that is too low can result in the gradient descent being slow, taking greater time to find the best accuracy. On the other hand, too high a learning rate can fail to find the best accuracy at all. Figure 7 shows how the accuracy of the XGB classifier improved as the learning rate was decreased. The accuracy peaked at approximately 77%, with a learning rate of .01. Adjusting this parameter alone allowed for an accuracy increase of approximately 4 percent. While it is not a drastic amount, the difference cannot be discounted either.

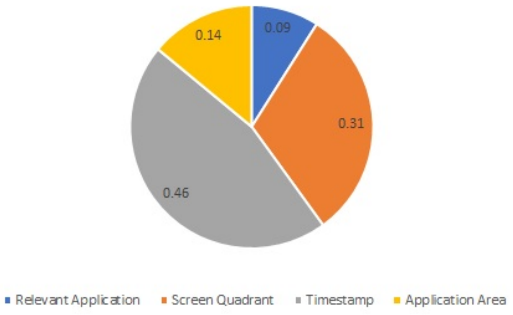

Figure 6: Feature Importance of XGB Model

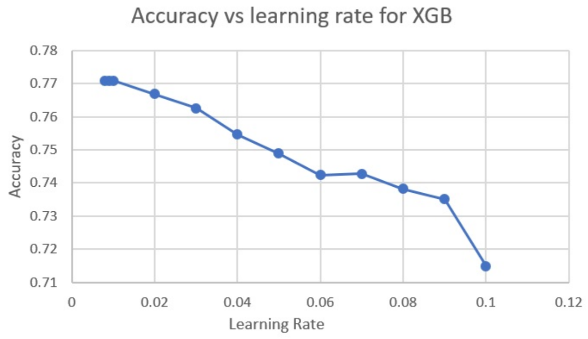

Figure 7: Accuracy of XGB Model with Adjustments to Learning Rate Parameter

5.4 Making Use of No Response Labels

A large number of the responses received were marked as '0', or Response. Additional models were created purely for testing an alternate theory. For these models, the category of 0/**No Response**, was combined into the 6-10/**High Attention** category. This was done under the assumption that students who responded '0' were in fact paying attention, and were responding as such merely to get the survey out of the way. It should be stressed that these models are inherently less reliable, as they are based off a premise that cannot be tested or is fully satisfactory.

As table 5 shows, the accuracy does increase with this change. XGB remains the strongest classifier for accuracy, achieving an accuracy of over 80%. As previously mentioned, these results exist only for theoretical testing. Due to the initial assumption made when creating these models, the results cannot be claimed to be trustworthy as a predictor. However, this alternate result does open up possibilities for future research. For example, if one could identify the reasoning behind a 'No response' answer, then that could lead to more accurate predictions overall.

Table 5: Results of Alternate Hypothesis accepting '0' as High Attention

Aggregate	RF	SVM	Adaboost	XGB
Accuracy	.73	.77	.74	.81

6 LIMITATIONS

While this research did achieve the results it was aiming for, there were some considerable limitations that complicated the experiment. One of the greatest issues encountered is that there is no way to ensure the ground truth received is 100% accurate. It is entirely possible for students to give false reports out of factors such as boredom or malicious intent. As a result, there is not guaranteed way of knowing from the self-reported scores what a student's real attention level is. Unfortunately, there is no way to remedy this problem, as it is an issue that lies with the individual rather than the experiment. The best solution is to make the assumption that all reports are given in good faith. s

Additionally, the Hawthorne Effect must always be taken into consideration [10]. This effect describes the phenomenon where subjects may alter their behavior, simply by being aware they are in an experiment. This change may not be a conscious decision, but it can still result in some differences from the person's natural behavior. While this effect can have an impact on a subject's affect, it is also impossible to measure without knowledge of the subjects' pre-experiment behavior.

The course material itself is also a limitation that must be considered. Students tended to use tools such as physical notebooks or calculators as opposed to the computer, somewhat limiting the responses received. This behavior should not directly impact their affect or the responses given. However, it does make them less inclined to use the computer or software, giving less data overall. This can be overcome by selecting a class that has a greater focus on computer-based learning.

7 CONCLUSION

The findings of this research do indeed show that eye gaze data is a useful feature for predicting attention. With a peak accuracy of approximately 77%, the machine learning models exceed a simple baseline model which only held an accuracy of 52%. While gaze data might necessarily be the best predictor on its own, it is able to work with other features to achieve a reliable level of accuracy. While the model may not be able to be created in real time, it stills offers an additional tool for any educator to use. Such a model allows them to look back on a class and determine when and how affect began to change drastically, in turn giving the professor the option to change future classes in response.

There are many avenues that can be taken to further this research. Perhaps the most straightforward option to take is to add more features. For example, the ability to track and record the professor's behavior would be an excellent feature to include. Such actions would certainly have a great, if not the largest, impact on a student's attention. Additionally, software can be developed to to automatically notify the student if a sufficient amount of time is detected as **Not attentive**. While this would be a more intrusive approach than the current research, it could also lead to better learning gains for the involved students. Just as students learn from their instructors to gain more knowledge, so too can this research help educators to better understand and help their students.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation under grant DUE-1525208 and the Georgia Southern University Faculty Research Committee. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the Georgia Southern University Faculty Research Committee.

REFERENCES

[1] R. Azevedo, A. Witherspoon, A. Chauncey, C. Burkett, and A. Fike. 2009. MetaTutor: A MetaCognitive Tool for Enhancing Self-Regulated Learning. In *2009 AAAI Fall Symposium Series*. Arlington, USA.

[2] R. Bixler and S. D'Mello. 2016. Automatic Gaze-based User-independent Detection of Mind Wandering during Computerized Reading. *User Modeling and User-Adapted Interaction* 26, 1 (2016), pp. 33–68.

[3] C. Conati and C. Merten. 2007. Eye-tracking for User Modeling in Exploratory Learning Environments: An Empirical Evaluation. *Knowledge-Based Systems* 20, 6 (2007), pp. 557–574.

[4] S. Craig, A. Graesser, J. Sullins, and B. Gholson. 2004. Affect and Learning: An Exploratory Look into the Role of Affect in Learning with AutoTutor. *Journal of educational media* 29, 3 (2004), pp. 241–250.

[5] S D'Mello, A. Olney, C. Williams, and P. Hays. 2012. Gaze Tutor: A Gaze-reactive Intelligent Tutoring System. *International Journal of human-computer studies* 70, 5 (2012), pp. 377–398.

[6] A. Graesser, P. Chipman, B. Haynes, and A. Olney. 2005. AutoTutor: An Intelligent Tutoring System with Mixed-initiative Dialogue. *IEEE Transactions on Education* 48, 4 (2005), pp. 612–618.

[7] N. Jaques, C. Conati, J. Harley, and R. Azevedo. 2014. Predicting Affect from Gaze Data during Interaction with an Intelligent Tutoring System. In *International Conference on Intelligent Tutoring Systems*. Springer, Honolulu, USA, pp. 29–38.

[8] B. Kort, B. Reilly, and R. Picard. 2001. An Affective Model of Interplay between Emotions and Learning: Reengineering Educational Pedagogy - Building A Learning Companion. In *Advanced Learning Technologies, 2001. Proceedings. IEEE International Conference on*. IEEE, Madison, USA, pp. 43–46.

[9] B. Lehman, M. Matthews, S. D'Mello, and N. Person. 2008. What Are You Feeling? Investigating Student Affective States During Expert Human Tutoring Sessions. In *International Conference on Intelligent Tutoring Systems*. Springer, Montreal, Canada, pp. 50–59.

[10] J. McCambridge, J. Witton, and D. Elbourne. 2014. Systematic Review of the Hawthorne Effect: New Concepts Are Needed to Study Research Participation Effects. *Journal of clinical epidemiology* 67, 3 (2014), pp. 267–277.

[11] K. Muldner, R. Christopherson, R. Atkinson, and W. Burleson. 2009. Investigating the Utility of Eye-tracking Information on Affect and Reasoning for User Modeling. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer, Trento, Italy, pp. 138–149.

[12] D. Rosengrant, D. Hearrington, K. Alvarado, and D. Keeble. 2012. Following Student Gaze Patterns in Physical Science Lectures. In *AIP Conference Proceedings*, Vol. 1413. AIP, Philadelphia, USA, pp. 323–326.

[13] K. Sharma, H. Alavi, P. Jermann, and P. Dillenbourg. 2016. A Gaze-based Learning Analytics Model: In-video Visual Feedback to Improve Learner's Attention in MOOCs. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*. ACM, Edinburgh, UK, pp. 417–421.

[14] J. Smallwood, D. Fishman, and J. Schooler. 2007. Counting the Cost of an Absent Mind: Mind Wandering as an Underrecognized Influence on Educational Performance. *Psychonomic bulletin & review* 14, 2 (2007), pp. 230–236.

[15] J. Zaletelj and A. Košir. 2017. Predicting Students' Attention in the Classroom from Kinect Facial and Body Features. *EURASIP Journal on Image and Video Processing* 2017, 1 (2017), 80.