# The Eyes Have It: Gaze-based Detection of Mind Wandering during Learning with an Intelligent Tutoring System

Stephen Hutt, Caitlin Mills, Shelby White, Patrick J. Donnelly, & Sidney K. D’Mello
University of Notre Dame
384 Fitzpatrick Hall, Notre Dame, IN, 46556, USA
{shutt, cmills4, swhite16, pdonnel4, sdmello}@nd.edu

## ABSTRACT

Mind wandering (MW) is a ubiquitous phenomenon characterized by an unintentional shift in attention from task-related to task-unrelated thoughts. MW is frequent during learning and negatively correlates with learning outcomes. Therefore, the next generation of intelligent learning technologies should benefit from mechanisms that detect and combat MW. As an initial step in this direction, we used eye-gaze and contextual information (e.g., time into session) to build an automated MW detector as students interact with GuruTutor – an intelligent tutoring system (ITS) for biology. Students self-reported MW by responding to pseudorandom thought-probes during the tutoring session while a consumer-grade eye tracker monitored their eye movements. We used supervised machine learning techniques to discriminate between positive and negative responses to the probes in a student-independent fashion. Our best results for detecting MW ($F_1$ of 0.49) were obtained with an evolutionary approach to develop topologies for neural network classifiers. These outperformed standard classifiers ($F_1$ of 0.43 with a Bayes net) and a chance baseline ($F_1$ of 0.19). We discuss our results in the context of integrating MW detection into an attention-aware version of GuruTutor.

## Keywords

eye-gaze, intelligent tutoring systems, mind wandering, attention-aware learning

## 1. INTRODUCTION

It is safe to say that most of us have had the experience of reading a text or listening to a lecture and then suddenly realizing that our thoughts have drifted to completely unrelated things, such as an upcoming vacation. This phenomenon, known as mind wandering (MW), refers to the unintentional shift of attention away from the current task towards internal task-unrelated thoughts [32]. MW is a ubiquitous phenomenon, estimated to occur as much as 50% of the time depending on the individual, task, and environment [16].

Not only does MW occur frequently, it can have detrimental influences on performance, especially during educational activities. Indeed, a recent meta-analysis revealed a negative correlation between MW and performance across a variety of tasks, such as lower recall in memory tasks and poor comprehension in reading tasks [24]. It is prudent to point out that MW is not always harmful and the tendency to day-dream has been shown to aid in certain types of tasks, such as creative problem solving [20]. However, research consistently shows that MW impairs performance in tasks requiring concentrated attentional focus and integration of information from the external environment as is the case with many learning activities [21].

Considering the negative influence of mind wandering on learning [27, 29, 30], it is important to take steps towards developing intelligent systems that help reorient attention to assuage the negative effects of MW. This requires an ability to monitor the locus of attention, detect students’ current attentional state, and provide a stimulus to direct focus back to the learning task [10]. Detecting MW is no easy task however. Although MW is related to other forms of disengagement, such as boredom, behavioral disengagement, and off-task behaviors [1, 2, 9, 18, 36], it is inherently distinct because it involves internal thoughts rather than overt expressive behaviors. This raises two challenges. First, while other disengaged behaviors often involve detectable behavioral markers (e.g., yawns signaling boredom), mind wandering is an internal state that can look similar to on-task states. Secondly, the onset and duration of MW cannot be precisely measured because MW can occur outside of conscious awareness.

Despite these challenges, there has been some progress toward automatic detection of mind wandering during reading (discussed as related works in Section 1.1). However, almost all of the current MW detectors focus on reading tasks, so their effectiveness is unclear during complex interactive tasks, such as learning with advanced learning technologies. Here, we explore for the first time, automated approaches for MW detection during learning with intelligent tutoring systems (ITS).

### 1.1 Related Work

In an early study attempting to detect MW in the context of learning [11], students were asked to read a paragraph about biology aloud, followed by either self-explanation or paraphrasing. Students self-reported how frequently they zoned out on a scale from 1 (all the time) to 7 (not at all). A supervised machine learning model trained on acoustic-prosodic features to classify low (1-3 on the scale) and high (5-7 on the scale) zone outs achieved an accuracy of 64%. However, it is unclear whether this detector could generalize to new students as the validation method did not ensure student-level independence across training and testing sets.

Some researchers have built MW detectors based on information readily available in log files collected during the reading (e.g., reading time, complexity of the text). For example, [19], attempted to classify whether students were MW while reading a screen of text using reading behaviors and features of the text, such as text difficulty. They were able to classify MW at 21% greater than chance using a leave-one-subject out cross-validation method. Similarly, another study [12] also attempted to predict MW during reading using textual features, such as word familiarity, difficulty, and reading time. However, rather than using supervised machine learning, they used a set of researcher-defined thresholds to ascertain if participants were "mindlessly reading" based on difficulty and reading time.

More recent studies have explored additional techniques to detect MW during self-paced computerized reading [5, 7, 12, 19]. In these studies, MW was measured via thought probes that occurred on pseudo-random screens (i.e. screen of text similar to a page of text). Participants responded either "yes" or "no" based on whether they were MW at the time of the probe. Supervised classification models were trained to discriminate the two responses using physiological features (e.g., skin conductance, temperature) [7] or eye-gaze [9], achieving accuracies ranging from 18% to 23% above chance and validated in a manner that generalized to new students. Further, combining the two modalities led to a 11% improvement in detection accuracy above the best individual modality [3].

Previous attempts to detect MW from eye-gaze are of particular relevance to the current paper. Eye tracking offers a unique possibility to automatically detect MW due to well-known relationships between visual attention and eye-movements. For example, MW has been associated with longer fixation durations [26] and more blinking in reading [33]. These and other relationships have been leveraged to build MW detectors during reading [4, 6] with moderate levels of success. However, it is unclear if these findings and corresponding detectors generalize to other activities, particularly activities where eye-gaze does not have the predictable patterns found in reading text.

## 1.2 Current Study and Novelty

The primary focus of this paper is to detect MW during learning with an ITS called GuruTutor. Previous work suggests that MW occurs, on average, once every two minutes during interactions with GuruTutor and is negatively correlated with learning gains [17], highlighting the importance of detecting MW in this context.

There are a number of novel aspects with this work. First, we study MW detection in an interactive context– an ITS with conversational dialogues and other embedded activities. Detection of MW during interactions with an ITS provides additional challenges compared to reading. In reading tasks, it is generally clear where the reader should be looking if they are engaged in the task and the eyes move across the screen in a predictable manner. However, in complex environments such as an ITS, there are far more paths the eyes may take, resulting in fewer predictable patterns, rendering MW detection more difficult.

Second, GuruTutor includes multiple activities, such as lecturing, scaffolded dialogue, concept mapping, and Cloze task completion. Each has a different visual layout, level of interactivity, and learning goal, presumably engendering different gaze patterns and levels of MW. By requiring our MW detector to work across a range of activities, we hope to have a solution that will generalize to additional learning technologies that may support quite different activity types.

Third, while researchers have typically used standard classification algorithms (e.g., Naïve Bayes, decision trees), we explore the use of a genetic algorithm (GA) to evolve neural networks (both topologies and connection weights) for detecting MW. This approach evolves the weights and topology concurrently, thereby implicitly integrating feature selection and feature weighting. Further, MW detection suffers from a data-imbalance problem in that the standard classifiers are skewed towards predicting the majority class, which is typically the class associated with Not MW. We address this issue by considering various GA fitness functions that focus on balancing the precision and recall of the minority MW class.

Fourth, we use a low-cost consumer-grade eye tracker to collect gaze data from participants as they interact with Guru. Research grade eye trackers can cost upwards of $40,000, so the use of affordable equipment (less than $150) increases the scalability of the detector for eventual deployment in real world learning environments such as computer-enabled classrooms.

# 2. DATA COLLECTION

We adopted a supervised classification approach for MW detection, which entailed collection of training and validation data.

## 2.1 Participants

Participants were 105 undergraduate students (69.5% female, average age 19.14) from a mid-sized, private university in the Midwest. Participants received extra credit or course credit for participating in the study.

## 2.2 GuruTutor

GuruTutor (Guru) is an ITS designed to teach biology topics through collaborative conversations in natural language. It is modeled after interactions with expert human tutors [22]. Guru engages the student through natural language conversations with an animated tutor agent that references a multimedia workspace, animating content relevant to the conversation (see Figure 1). Students type in responses in a conversational style that Guru analyzes using natural language processing. Guru maintains a student model which it uses to tailor instruction to individual students. Guru has been shown to be effective at promoting learning and retention at levels similar to human tutors [22].

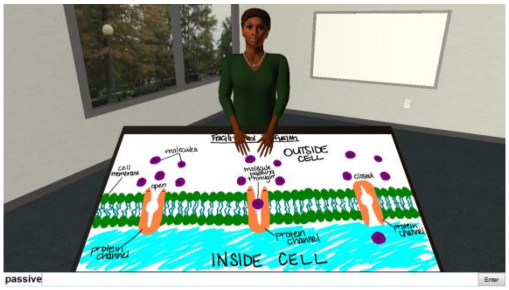

**Figure 1. Example of Guru during CGB Phase**

Guru presents biology topics aligned with state curriculum standards (e.g., *cellular respiration*), typically lasting between 15 to 40 minutes each. Each topic contains sets of interrelated concepts and facts (e.g., *proteins help cells regulate functions*). Guru begins each new topic with a brief preview to introduce it to the student, followed by a five phase session that encourages students to build and articulate their understanding of the concepts. These five phases are described below.

**Common-Ground-Building Instruction (CGB Instruction).** Biology lessons often involve specialized terminology that needs to be well understood before it is possible to move on to more collaborative knowledge building activities. Therefore, Guru begins with a collaborative lecture phase that covers basic information and terminology relevant to the topic. **Intermittent Summaries (Summary).** Following CGB, students generate summaries using natural-language to describe the content covered. These summaries are automatically analyzed to determine which concepts to target throughout the remainder of the session. **Concept Maps**. For the target concepts, students complete skeleton concept maps, node-link structures that are automatically generated from concept text. **Scaffolded Dialogue.** Next students complete a scaffolded natural language dialogue in which GuruTutor uses a Prompt → Feedback → Verification Question → Feedback → Elaboration cycle to cover target concepts. If a student shows difficulty mastering particular concepts, a second Concept Maps phase is initiated followed by an additional Scaffolded Dialogue phase. **Cloze Task.** The session concludes with a cloze task requiring students to complete an ideal summary of the topic by filling in blanks to connect key words to related concepts.

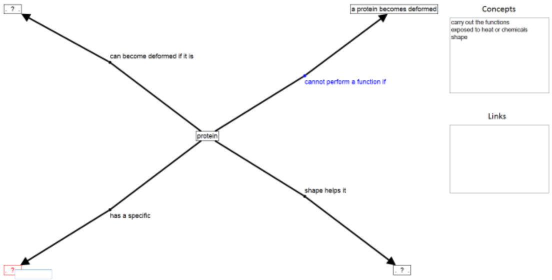
?
a protein becomes deformed
Concepts
carry out the functions
exposed to heat or chemicals
shape
can become deformed if it is
cannot perform a function if
protein
Links
shape helps it
has a specific
?
?

**Figure 2. Example of Guru during Concept Maps**

## 2.3 Procedure

All experimental procedures were reviewed and approved by the university's ethics board. After signing an informed consent, participants were seated at a desk in front of a 15-inch laptop. A Tobii EyeX eye-tracker was positioned directly under the laptop screen using a magnetic strip based on the guidelines provided by Tobii.

Participants were asked to sit comfortably with the chair pulled up to the desk. Next, participants were given an explanation of MW and were given detailed instructions for how to respond to the mind wandering probes (see below) during learning with Guru. Specifically, MW was defined as "when you realize that you are no longer paying attention to what you're supposed to be doing, for example, instead of thinking about the biology, you may be thinking about something else altogether."

After receiving initial instructions, a 60 second calibration process occurred before beginning the learning session. Participants were dynamically instructed about their seating and head position in order for the eye tracker to pick up their eye gaze.

Then, one of six biology topics from Guru was assigned to each participant: Interphase, Osmosis, Biochemical Catalysts, Carbohydrate Function, Protein Function, or Facilitated Diffusion. Following a pretest on the assigned topic, participants began the Guru tutoring session. Afterwards, participants completed a posttest and were fully debriefed.

## 2.4 Mind Wandering Probes

Mind wandering was measured during learning with Guru using auditory thought probes, which is a standard approach in the literature [31]. Participants were probed at pseudo-random intervals with probes occurring every 90-120 seconds, this was based on previous work investigating how often MW occurs[17]. If the tutor was speaking at the time the probe was triggered, the probe was paused until the tutor finished speaking so as to not interrupt the conversation flow. Probes consisted of an auditory beep that automatically paused the tutoring session. An opaque overlay would then appear on screen, instructing the participant to press the "N" key if they were not mind wandering, the "I" key if they were intentionally (deliberately) mind wandering, or the "U" key if they were unintentionally (spontaneously) mind wandering. In this study, we do not differentiate between intentional and unintentional mind wandering, and "I" and "U" responses were coded as "MW" to indicate mind wandering occurred. Participants encountered an average of ten probes over the course of the session. We obtained a total of 1104 reports to thought probes, 17% of which corresponded to episodes of MW.

# 3. MODEL BUILDING

Supervised machine learning models were built to detect MW using eye-gaze data and contextual information from Guru.

## 3.1 Feature Engineering

We calculated features from a short window of time preceding each auditory probe, exploring window sizes ranging from 3 to 30 seconds. We did not consider windows shorter than 3 seconds, as they most often did not contain sufficient gaze data. We discarded windows where not all the eye-gaze features could be computed, such as cases when the face was occluded or the student was looking down at the keyboard. For the smallest window (three seconds) 418 instances were removed, lowering the MW rate to 15.5%. A total of 156 instances were removed for all other window sizes, leaving the average MW rate unaffected (17%).

**Table 1. Eye-gaze features**

| | |
|---|---|
| **Fixation Duration** | duration in milliseconds of fixation |
| **Saccade Duration** | duration in milliseconds of saccade |
| **Saccade Length** | distance of saccade |
| **Saccade Angle Absolute** | angle in degrees between the x-axis and the saccade |
| **Saccade Angle Relative** | angle of the saccade relative to previous gaze data. |
| **Saccade Velocity** | Saccade Length / Saccade Duration |
| Fixation Dispersion | root mean square of the distances from each fixation to the average fixation position in the window |
| Horizontal Saccade Proportion | proportion of saccades with angles no more than 30 degrees above or below the horizontal axis |
| Fixation Saccade Ratio | ratio of Fixation Duration to Saccade Duration |

*Note.* Bolded cell indicates that the total number, mean, median, min, max, standard deviation, range, kurtosis, and skew of the distribution of each measurement were used as features.

**Gaze Features**. Eye movements are measured by fixations (i.e. points in which the gaze was maintained on the same location) and saccades (i.e. the movement of the eyes between fixations). We calculated fixations and saccades from the raw eye-gaze data using the Open Gaze and Mouse Analyzer (OGAMA) [35], an open source package for eye tracking analysis. Next, gaze features were computed for each from the fixations and saccades (see Table 1) in that window. We considered six general measures based on fixations and saccades. For these gaze measures, we calculated the number, mean, median, min, max, standard deviation, range, kurtosis, and skew of the distributions of each measure across the time window, yielding 54 features. We also included three other features (listed in Table 1), yielding a total of 57 gaze features.

**Contextual Features**. The gaze features were complemented with eight contextual features that provide a snapshot of the student-tutor interaction context during each window. One feature was the assigned biology *topic*. A second encoded participants' *pretest* scores on that topic. The next three of these features describe participants' progress within Guru, such as the *current phase* of the session (e.g., cloze, concept map, etc.), the amount of elapsed *time into the session*, and the amount of elapsed *time into the current phase*. The last three context features focused on participants' overall interaction with Guru, measured by the amount of positive, neutral, and negative feedback received.

## 3.2 Addressing Class Label Imbalance

Only 17% of the 1104 thought probes were reports of MW, thereby leading to substantial data skew. This imbalance between the class labels poses a challenge as some supervised learning methods tend to bias predications towards the majority class label. To compensate for this concern, synthetic oversampling was applied to provide a more balanced class distribution on the training set only. The SMOTE algorithm [8] creates synthetic instances of the minority class by interpolating feature values between an instance and randomly chosen nearest neighbors. No SMOTING was done on the testing set in order to ensure validity of the predictions.

## 3.3 Classification Models

We evaluated five classifiers frequently explored for the detection of MW [6, 7]. These included Bayesian networks, logistic regression classifiers, multilayer perceptrons (MLP), random forests, and support vector machines (SVM) using implementations from the WEKA data mining software [14].

We also considered a neural network trained using a genetic algorithm (GA), which is a type of evolutionary algorithm for optimization and search problems that uses techniques loosely inspired by biological natural selection. GAs maintain a population of candidate solutions (phenotypes), each with a set of properties (genotypes). These individual solutions evolve over time guided by a fitness function. At each generation, the fitness function is used to rank the candidate solutions, allowing elimination of inferior solutions and selection of the best candidates to the new generation. New candidate solutions are created at each generation through the mechanisms of mutation, a pseudo-random perturbation of an individual's genotype, and cross-over, the combination of aspects of the genotypes of multiple fit individuals.

**NEAT Algorithm.** In this study, we used a GA to evolve an artificial neural network for MW detection. We used the NeuroEvolution of Augmenting Topologies (NEAT) algorithm to evolve the topology of neural network alongside an evolution of the network weights [34]. Because NEAT evolves both the weights and topology of the network, it must implement the genetic operators of mutation and crossover in a unique way to handle differences between network topologies. NEAT uses population speciation to track individuals with similar topologies, restricting crossover to individuals with similar network topologies to ensure the resulting new topology is coherent. Mutation of the topology occurs in two ways, either by the creation of a hidden node or the addition or removal of a link between nodes. As the size of the networks may grow larger in each new generation, constraints are imposed to penalize large networks that exceed a complexity threshold.

To encourage innovation in new generations, NEAT implements speciation by grouping networks that share similar topologies into the same population. The populations are determined by a distance metric that computes the distance of a topology of an individual from the initial topology of the species. New populations are created as new networks that are dissimilar from any existing population evolve. This strategy allows the generation of new individuals by applying genetic operators on similar individuals in order to maintain viable network topologies without hindering the ability of the GA to develop new and unique networks.

**Using NEAT for MW Detection.** We used SharpNeat, a popular implementation of the NEAT algorithm in the C# language [28]. We tuned the evolution variables on our data in preliminary experiments. We used a population of 150 individuals and ran the algorithm for 500 generations. We also determined a complexity threshold to prune overly complex networks. Because evolutionary algorithms are non-deterministic, we ran these classifiers over multiple iterations in each experiment.

The effectiveness of an evolutionary algorithm depends on the evaluation of individuals using the fitness function. We considered three different fitness functions that were informed by [13]. The first function evaluates candidate networks using the overall accuracy (recognition rate) of the model. The second function evaluates the networks considering the $F_1$ measure for the class label of interest, which in our case is MW (denoted as $F_1$-MW). The third evaluates the networks using the Youden's J-statistic, (a variation on Cohen's Kappa, sometimes called "informedness" [23]) which is defined as *sensitivity + specificity – 1* of MW.

## 3.4 Cross-Validation

All experiments were conducted using leave-several-participants-out cross-validation. For each iteration of the classifier, instances from 66% of the participants were assigned to a training set and the remaining instances of the other 33% participants were assigned to a test set. This process ensures that no instances of any individual participant could appear in both the training and test sets within a fold. This process was repeated for 15 folds, and the results accumulated. We selected 15 iterations in order to balance time taken to build the models (as evolutionary approaches are slow) and reliability by testing multiple training/testing set pairs. Minority oversampling (SMOTING) occurred within each fold and on the training set only.

# 4. RESULTS

We report the $F_1$ measure in our evaluation of our results. This measure is common in information retrieval tasks and provides a balance between precision and recall. Because our intention is to detect instance of MW, we focus on the $F_1$ score of the MW label as our key metric. This is a very strict evaluation criterion as the base rate of MW is only 17% in our data. To facilitate comparisons with previous (and future work), we also reported the $F_1$ score for the majority Not MW class (83% of instances), as well as the weighted $F_1$ score.

## 4.1 Comparing Window Size

In our first experiment, we explored the influence of various window sizes ranging from 3 to 30 seconds. As we are interested in general trends, we average results of the five standard classifiers and the three NEAT classifiers. (see Figure 3). These results illustrate a general trend of improved performance for the larger windows, although these differences may not be overly large. In the remainder of this work, we considered a 30 second window in our experiments as it generally resulted in the highest $F_1$ scores.

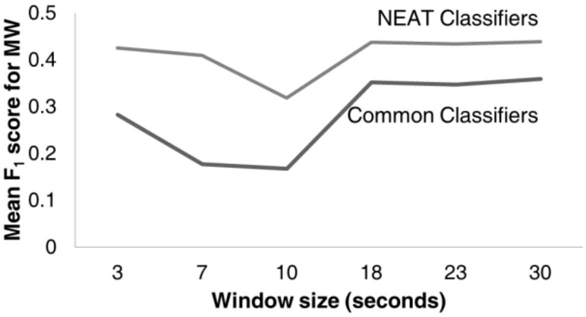

**Figure 3. Comparison of different window sizes.**

## 4.2 Comparison of Classifiers

In Table 3 we report the results of the classifiers considering a 30 second analysis window, informed by our experiment in Section 4.1. The highest $F_1$ for MW is denoted in bold for both the common classifiers and NEAT implementations that varied by fitness function. For comparison, a chance-level baseline was created by randomly assigning a class label to each instance based on the observed MW rate of 17%. We note that all of the classifiers showed an improvement in detecting the target minority class of MW over the chance model.

**Table 2. MW detection results by classifier for 30 second window**

| | $F_1$ of MW | $F_1$ of Not MW | Overall $F_1$ |
|---|---|---|---|
| **Standard Classifiers** | | | |
| Bayesian Network | **0.43** | **0.73** | 0.68 |
| Logistic Regression | 0.38 | 0.79 | 0.72 |
| MLP | 0.30 | 0.83 | 0.74 |
| SVM | 0.37 | 0.76 | 0.70 |
| Random Forest | 0.23 | 0.86 | 0.75 |
| **NEAT Classifiers** | | | |
| Fitness: Accuracy | 0.36 | 0.76 | 0.69 |
| Fitness: $F_1$-MW | **0.49** | **0.58** | 0.57 |
| Fitness: Youden J | 0.44 | 0.69 | 0.65 |
| **Baseline** | 0.19 | 0.83 | 0.73 |

Among the common classifiers, Bayesian network achieved the highest $F_1$ score for MW. This was also the case in previous MW eye-gaze detectors in other domains [6]. The overall $F_1$ score for the Bayesian network was lower than for other classifiers, ostensibly because the other classifiers tend to over predict the majority class. For NEAT, using the $F_1$-MW score as the fitness function resulted in the overall best $F_1$ score for MW. NEAT with Youden's J- statistic as the fitness function did yield a slightly more balanced detector with an increase in $F_1$ of Not MW. Importantly, the best NEAT classifier outperformed the Bayesian network at detecting MW, which is our target class of interest. In Table 3 we show the confusion matrices for the three classifiers that obtained the highest $F_1$ score for MW: the Bayesian network, NEAT-$F_1$-MW, and NEAT-Youden. NEAT-$F_1$-MW yielded a substantially higher hit rate than the other two classifiers, but also suffered from a high false positive (FP) rate. The Bayesian network and NEAT-Youden had similar patterns of errors in that they had both lower hit rates as well as FP rates. Based on these results, we consider NEAT-$F_1$-MW and the Bayesian network in subsequent analyses.

**Table 3. Confusion matrices for the three best classifiers**

| **Actual** | **Predicted** | |
|---|---|---|
| *Bayes Net* | MW | Not MW |
| MW | 0.52 (hit) | 0.48 (miss) |
| Not MW | 0.34 (false pos.) | 0.66 (correct rej.) |
| *NEAT-$F_1$-MW* | MW | Not MW |
| MW | 0.69 (hit) | 0.31 (miss) |
| Not MW | 0.54 (false pos.) | 0.46 (correct rej.) |
| *NEAT-Youden* | MW | Not MW |
| MW | 0.55 (hit) | 0.45 (miss) |
| Not MW | 0.41(false pos.) | 0.59 (correct rej.) |

## 4.3 Gaze only vs. Gaze + Context Features

We investigated the utility of contextual features over the gaze features alone (see Table 4). The addition of contextual features improved the $F_1$ score for the minority class of MW for NEAT and correspondingly for the majority Not MW class for the Bayesian network. Overall, the improvements in performance were small, suggesting that the gaze features were more important to the detection of MW compared to the contextual features.

**Table 4. Gaze (G) vs. Gaze + Context (G+C) features**

| Classifier | Feature | $F_1$ of MW | $F_1$ of Not MW | Overall $F_1$ |
|---|---|---|---|---|
| Bayesian network | G | 0.45 | **0.69** | 0.65 |
| | G+C | 0.43 | **0.73** | 0.68 |
| NEAT-$F_1$-MW | G | **0.44** | 0.58 | 0.56 |
| | G+C | **0.49** | 0.58 | 0.57 |

## 4.4 Oversampling vs. No Oversampling

In Section 3.2, we discussed the imbalance between instances of MW and Not MW in the dataset, and addressed this difficulty by supplementing the training data with the SMOTE oversampling technique. To study the effect of SMOTE, we compared the Bayesian network and the best NEAT classifier on datasets with and without these synthetic training instances (see Table 5). We confirmed that synthetic oversampling indeed improved the classification of the MW (the minority class) for NEAT at the cost of detecting the majority class. Thus, SMOTING played a critical role in reducing the tendency to over predict to the majority class. SMOTING had no notable effect for the Bayesian network, which seemed to be more impervious to data skew.

**Table 5. Results with and without oversampling.**

| Classifier | SMOTE | $F_1$ of MW | $F_1$ of Not MW | Ove all $F_1$ |
|---|---|---|---|---|
| Bayesian net | No | 0.41 | 0.75 | 0.70 |
| | Yes | 0.43 | 0.73 | 0.68 |
| NEAT-$F_1$-MW | No | **0.42** | **0.75** | 0.79 |
| | Yes | **0.49** | **0.58** | 0.57 |

## 4.5 Analysis of Features

Neural networks use a mathematical approach to transform and combine input features to useful output. Thus, we can learn more about the structure of our MW detector by investigating the topologies formed during the evolutionary process. For example, a network with a densely connected hidden layer would be performing a large amount of internal calculations compared a sparsely connected layer.

To better understand our MW detector's structure, we examined each of the 15 iterations of the *NEAT-$F_1$-MW* model and investigated the networks that survived to the final generation in each case. Across the networks the mean number of hidden nodes in the network is 1.6 (min 0, max 3), the average number of inputs actually used in the final network is 17.133 (min 8, max 36) and the average number of connections is 21.46 (min 9, max 44). The number of hidden nodes here is low, but considering the large number of inputs to a small number of outputs, this is to be expected. The algorithm also biases towards smaller networks to avoid bloat.

Developing neural network topologies also provides inherent feature selection that takes place as the network structures evolve to subsequent generations. This provides an opportunity to explore which features were most useful in detecting MW. Seven features appeared in at least half of the final networks as shown in Table 6.

**Table 6. Cohen's *d* of most commonly used features**

| Feature | Cohen's *d* |
|---:|---:|
| Fixation Duration Skew | -0.27 |
| Minimum Fixation Duration | 0.17 |
| Mean Saccade Duration | 0.32 |
| Saccade Duration Kurtosis | -0.16 |
| Saccade Duration Skew | -0.17 |
| Minimum Saccade Velocity | -0.15 |
| Fixation to Saccade Ratio | -0.17 |
| Pre Test Score | -0.18 |

We compared these seven features across the MW and not MW instances using an effect size measure (Cohen's *d*). An effect size measure is appropriate for this comparison in order to evaluate the direction and magnitude of the differences between the two classes. Positive values depict higher values for instances of MW (see Table 6). In general, the differences reported in this paper are consistent with previous work examining eye gaze surrounding MW episodes during reading [4]. Two of the seven features had differences across the MW and not MW classes consistent with small effect sizes ($|d| > .2$). The largest difference was seen for mean saccade duration ($d = .32$). This finding suggests that participants tend to have longer gaps between fixations leading up to a MW episode as opposed to more rapid eye movements between fixations. A similar effect size was found for fixation duration skew ($d = -.27$), which suggests that there is a higher probability that participants would have shorter fixations before a MW episode occurs compared to when their attention is on task.

It is important to point out that the low Cohen's *d* values ($< |.2|$) are not entirely surprising given the nature of neural networks. The network employs a combination of features and the combination sets that prove to be most effective for MW detection may not be consistent with the overall largest mean differences. Instead, the important thing to note is that these seven features were the most consistent across all iterations.

It is also worth mentioning that only one context feature was present in over half of the final networks: pre-test score. Instances of MW were associated with lower pre-test scores, indicating that when participants were more likely to mind wander if they did not understand the topic well to begin with.

# 5. GENERAL DISCUSSION

Mind wandering occurs frequently during learning and has a negative impact on learning outcomes [21]. An attention-aware learning technology [10] that can automatically detect MW could intervene to re-engage learners, assuaging the cost of MW on comprehension to improve learning. However, MW is a covert, internal state with no obvious behavioral markers, making it difficult to detect. Although strides have been made to detect MW in the context of self-paced reading, MW detection has not yet been attempted in the context of an ITS – a challenge we addressed in the current paper. In the remainder of this section, we discuss our main findings, consider potential applications, and discuss limitations and future work.

## 5.1 Main Findings

MW detection during reading tasks is supported by decades of research on MW and eye movements [25]. However, more complex learning interfaces, such as the ITS used here, are not afforded such predictable patterns of eye movements. Despite these challenges, we demonstrated the ability of a neural network trained using a GA to detect MW in the context of learning with an ITS. We were able to accurately classify MW with an $F_1$ of 0.49 at detecting the minority MW class. Although this result is modest, it is an important first step in detecting MW in this novel domain.

In most machine learning tasks, a large imbalance in the distribution of class labels results in a degraded performance at predicting the minority class label [15]. This is a major issue for MW detection as its rate of occurrence is around 20% to 40% in learning contexts [27] and in our case it was 17%. We addressed the data imbalance by using a synthetic oversampling technique and by tweaking the fitness function of the GA in order to help the classifiers in detecting the minority class of MW. We believe that this combined approach might be beneficial for other classification problems when there is severe data skew.

Since MW detection in the context of learning from an ITS is still in its infancy, it was important for us to adopt a method that will generalizable for future work in this area. The eye gaze feature set was limited to eye movements that were independent of the specific content being displayed on the screen. This enabled our models to operate across Guru's multiple instructional activities, each with very different visual displays.

In addition to the gaze features, a second set of features included the context of the learning session. A comparison of model performance with and without contextual features revealed that contextual features added a small, but not substantial, improvement in detection accuracy. This finding further illustrates the idea that eye gaze can be a powerful signal of attention, regardless of the learning context.

An analysis of the most consistent features in the model point to seven important features, six of which are gaze features. MW episodes had a longer mean saccade duration, yet smaller fixation duration skew. The longer mean saccade duration preceding MW is consistent with prior research, which suggests that MW signals a breakdown at very basic levels of perceptual processing [30] – in this case, being slower to direct your eyes form one point to another. Most of the effect sizes (*d*'s) reported are objectively small effects; however, we feel that obtaining a sense of consistent features and how the relate to MW is a major contribution at this stage in the of MW detection.

All data was collected using low-cost, consumer-grade eye trackers (less than $150). This is a marked contrast compared to many research-grade trackers that can cost tens of thousands of dollars. Our goal is eventual deployment of our models at scale, thereby allowing us to test generalizability in more diverse contexts. For this reason, it was important to ensure that our models were validated in a student-independent manner, which increases our models' ability to generalize to new students. Taken together, these results increase our confidence that the models will generalize more broadly, though this claim requires further empirical validation.

## 5.2 Applications

The key application of this work is to develop an attention-aware version of Guru that detects and combats MW in real-time. Once the goal of MW detection is realized, Guru has a number of paths to pursue to re-engage attention.

At an immediate level, one initial effect of MW is that the student simply fails to attend to a unit of information or a salient event in the learning environment. The unattended information, question, or event is needed to construct an adequate mental model so that subsequent knowledge can be assimilated or the student will be left behind. Thus, a simple direct approach is to reassert the missed information ("e.g., Mary, let me repeat that…..") or highlight the information by directing attention to specific areas of the display (e.g., "Mary, you might want to look at the highlighted image showing the chromosomes duplicating"). Taking a somewhat different approach, Guru can also launch a sub-dialogue where it asks a content-specific question (e.g., "Mary, what happens to the chromosomes when they duplicate") or asks the student to complete a mini-activity (e.g., "Mary, we now have a simulation of the first phase in mitosis. Can you…."). Guru can also ask the student to self-explain when MW is detected.

Additional measures might be needed if MW persists despite these intervention strategies. One option is to simply change to a new activity. Guru might even suggest changing topics or offering a choice for what students would like to do next. If all else fails, Guru might even suggest that the student take a break.

It is important to note that the proposed intervention strategies rely on MW detection, which is inherently imperfect. The detector might inaccurately assert that a student is MW when they are not (false alarms) or it might assert that a student is actively attending when they are in fact MW (misses). MW detection does not need to be perfect as long as we account for this in MW interventions. For example, Guru can adopt a probabilistic approach where the MW detector provides an estimate of the likelihood that the student is MW. This likelihood will guide whether an intervention is launched (i.e. if the likelihood of MW is 70%, there is a 70% chance that an intervention will be triggered). Second, interventions can be designed to be "fail-soft" in that there are no harmful effects if delivered incorrectly.

## 5.3 Limitations and Future Work

There were several limitations with this study. One key limitation pertains to the moderate MW detection accuracy. Although, we detected MW above chance levels using several different classifiers, these results leave room for improvement. Ongoing work seeks to reduce the false positive rate while increasing the hit rate for our MW models by expanding our feature set and incorporating temporal information in the machine learning.

We designed our approach to include a low-cost eye tracker, however, these consumer models have a lower sampling-rate, limiting the accuracy of the eye-gaze data compared to research-grade eye trackers. Furthermore, although we desire to eventually deploy our system in noisy classroom environments, we only tested our system in a quiet lab setting.

This work is also limited by the features used in the supervised learning process, which were a small and potentially restrictive subset of gaze features. We also did not model temporal patterns of eye movements, such as examining if the participant revisited an area of the screen they had previously viewed. Additionally, we only used a small number of contextual features. Future work may consider a utilizing log files from the tutoring session more extensively to create more in-depth context features (e.g., content, timing, and length of student responses, etc.).

The results of this study invite several avenues for improvement which we will explore as future work. First, we will explore additional eye-gaze features, such as those that track localized regions of interest but at a level of abstraction that does not limit generalizability to additional interfaces. Informed by our observation that the inclusion of contextual features improved detection of MW, we will explore additional contextual features from the ITS, again with an eye for more generalizable features (e.g., response time). Furthermore, it is possible to build multiple MW detector specialized for different phases in the Guru tutoring sessions, although this would require a large amount of data and would make these detectors less able to generalize to other ITSs. Finally, we will collect data in the real-world context of a computer-enabled classroom where 20-30 students interact with Guru on individual computers while their gaze is being tracked. Indeed, preliminary data collection on this front is already underway.

## 5.4 Concluding Remarks

Attention is a crucial part of learning. An attention-aware ITS that can detect a student's attentional state as well as redirect their attention to better engage them in the learning task could be very beneficial for engagement and learning. Attention-awareness, however, requires monitoring of attention, which has historically been limited to the lab. However, advances in consumer-grade eye-tracking have opened up the possibility of gaze tracking during learning with ITSs and other technologies, thereby enabling a new generation of attention-aware cyberlearning.

# 6. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (NSF) (DRL 1235958 and IIS 1523091). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

# 7. REFERENCES

[1] Arroyo, I. et al. 2007. Repairing disengagement with non-invasive interventions. *AIED* (2007), 195–202.

[2] Baker, R.S.J. 2007. Modeling and understanding students' off-task behavior in intelligent tutoring systems. *Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), 1059–1068.

[3] Bixler, R. et al. 2015. Automatic Detection of Mind Wandering During Reading Using Gaze and Physiology. *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction* (New York, NY, USA, 2015), 299–306.

[4] Bixler, R. and D'Mello, S. 2015. Automatic Gaze-Based Detection of Mind Wandering with Metacognitive Awareness. *User Modeling, Adaptation and Personalization: 23rd International Conference, UMAP 2015, Dublin, Ireland, June 29 -- July 3, 2015. Proceedings*. F. Ricci et al., eds. Springer International Publishing. 31–43.

[5] Bixler, R. and D'Mello, S. 2015. Automatic gaze-based user-independent detection of mind wandering during computerized reading. *User Modeling and User-Adapted Interaction*. (2015), 1–36.

[6] Bixler, R. and D'Mello, S. 2014. Toward Fully Automated Person-Independent Detection of Mind Wandering. *User Modeling, Adaptation, and Personalization*. V. Dimitrova et al., eds. Springer International Publishing. 37–48.

[7] Blanchard, N. et al. 2014. Automated Physiological-Based Detection of Mind Wandering during Learning. *Intelligent Tutoring Systems*. S. Trausan-Matu et al., eds. Springer International Publishing. 55–60.

[8] Chawla, N.V. et al. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence research*. (2002), 321–357.

[9] Cocea, M. and Weibelzahl, S. 2011. Disengagement Detection in Online Learning: Validation Studies and Perspectives. *IEEE Transactions on Learning Technologies*. 4, 2 (Apr. 2011), 114–124.

[10] D'Mello, S.K. 2016. Giving Eyesight to the Blind: Towards Attention-Aware AIED. *International Journal of Artificial Intelligence in Education*. (2016), 1–15.

[11] Drummond, J. and Litman, D. 2010. In the Zone: Towards Detecting Student Zoning Out Using Supervised Machine Learning. *Intelligent Tutoring Systems*. V. Aleven et al., eds. Springer Berlin Heidelberg. 306–308.

[12] Franklin, M.S. et al. 2011. Catching the mind in flight: using behavioral indices to detect mindless reading in real time. *Psychon Bull Rev*. 18, 5 (Oct. 2011), 992–997.

[13] Freitas, A.A. 2002. *Data mining and knowledge discovery with evolutionary algorithms*.

[14] Hall, M. et al. 2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*. 11, 1 (2009), 10–18.

[15] Jeni, L.A. et al. 2013. Facing Imbalanced Data–Recommendations for the Use of Performance Metrics. *Proceedings of the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction* (Washington, DC, USA, 2013), 245–251.

[16] Killingsworth, M.A. and Gilbert, D.T. 2010. A wandering mind is an unhappy mind. *Science*. 330, 6006 (2010), 932–932.

[17] Mills, C. et al. 2015. Mind Wandering During Learning with an Intelligent Tutoring System. *Artificial Intelligence in Education: 17th International Conference, AIED 2015, Madrid, Spain, June 22-26, 2015. Proceedings*. C. Conati et al., eds. Springer International Publishing. 267–276.

[18] Mills, C. et al. 2014. To Quit or Not to Quit: Predicting Future Behavioral Disengagement from Reading Patterns. *Intelligent Tutoring Systems* (2014), 19–28.

[19] Mills, C. and D'Mello, S. 2015. Toward a Real-time (Day) Dreamcatcher: Sensor-Free Detection of Mind Wandering During Online Reading. *Proceedings of the 8th International Conference on Educational Data Mining*. (2015).

[20] Mooneyham, B.W. and Schooler, J.W. 2013. The costs and benefits of mind-wandering: a review. *Can J Exp Psychol*. 67, 1 (Mar. 2013), 11–18.

[21] Olney, A.M. et al. In Press. Attention in Educational Contexts: The Role of the Learning Task in Guiding Attention. *The Handbook of Attention*. J. Fawcett et al., eds. MIT Press.

[22] Olney, A.M. et al. 2012. Guru: A Computer Tutor That Models Expert Human Tutors. *Intelligent Tutoring Systems*. S. Cerri et al., eds. Springer Berlin Heidelberg. 256–261.

[23] Powers, D.M.W. 2012. The Problem with Kappa. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (Stroudsburg, PA, USA, 2012), 345–355.

[24] Randall, J.G. et al. 2014. Mind-wandering, cognition, and performance: a theory-driven meta-analysis of attention regulation. *Psychol Bull*. 140, 6 (Nov. 2014), 1411–1431.

[25] Rayner, K. 1998. Eye movements in reading and information processing: 20 years of research. *Psychol Bull*. 124, 3 (Nov. 1998), 372–422.

[26] Reichle, E.D. et al. 2010. Eye movements during mindless reading. *Psychol Sci*. 21, 9 (Sep. 2010), 1300–1310.

[27] Risko, E.F. et al. 2012. Everyday attention: variation in mind wandering and memory in a lecture. *Applied Cognitive Psychology*. 26, 2 (2012), 234–242.

[28] SharpNEAT: 2016. *http://sharpneat.sourceforge.net/*. Accessed: 2016-02-22.

[29] Smallwood, J. et al. 2007. Counting the cost of an absent mind: Mind wandering as an underrecognized influence on educational performance. *Psychonomic Bulletin & Review*. 14, 2 (2007), 230–236.

[30] Smallwood, J. 2011. Mind-wandering while reading: Attentional decoupling, mindless reading and the cascade model of inattention. *Language and Linguistics Compass*. 5, 2 (2011), 63–77.

[31] Smallwood, J. et al. 2008. When attention matters: the curious incident of the wandering mind. *Mem Cognit*. 36, 6 (Sep. 2008), 1144–1150.

[32] Smallwood, J. and Schooler, J.W. 2006. The restless mind. *Psychol Bull*. 132, 6 (Nov. 2006), 946–958.

[33] Smilek, D. et al. 2010. Out of mind, out of sight: eye blinking as indicator and embodiment of mind wandering. *Psychological Science*. 21, 6 (Jun. 2010), 786–789.

[34] Stanley, K.O. and Miikkulainen, R. 2002. Evolving Neural Networks Through Augmenting Topologies. *Evolutionary Computation*. 10, 2 (2002), 99–127.

[35] Vosskuhler, A. et al. 2008. OGAMA (Open Gaze and Mouse Analyzer): open-source software designed to analyze eye and mouse movements in slideshow study designs. *Behav Res Methods*. 40, 4 (Nov. 2008), 1150–1162.

[36] Wixon, M. et al. 2012. WTF? detecting students who are conducting inquiry without thinking fastidiously. *User Modeling, Adaptation, and Personalization*. Springer. 286–296.