Using Machine Learning to Detect SMART Model Cognitive Operations in Mathematical Problem-Solving Process

Jiayi Zhang
University of Pennsylvania
joycez@upenn.edu

Ryan S. Baker
University of Pennsylvania
ryanshaunbaker@gmail.com

Caitlin Mills
University of Minnesota
caitlin.s.mills@gmail.com

Tyron Young
Oakwood School
tyron.young@gmail.com

Juliana Ma.
Alexandra L. Andres
University of Pennsylvania
aandres@gse.upenn.edu

Jaclyn Ocumpaugh
University of Pennsylvania
jlocumpaugh@gmail.com

Jamiella Brooks
University of Pennsylvania
brooksdj@upenn.edu

Stephen Hutt
University of Denver
Stephen.hutt@du.edu

Nidhi Nasiar
University of Pennsylvania
nasiar@upenn.edu

Sheela Sethuaman
CueThink
sheela@cuethink.com

Self-regulated learning (SRL) is a critical component of mathematics problem-solving. Students skilled in SRL are more likely to effectively set goals, search for information, and direct their attention and cognitive process so that they align their efforts with their objectives. An influential framework for SRL, the SMART model (Winne, 2017), proposes that five cognitive operations (i.e., searching, monitoring, assembling, rehearsing, and translating) play a key role in SRL. However, these categories encompass a wide range of behaviors, making measurement challenging – often involving observing individual students and recording their think-aloud activities or asking students to complete labor-intensive tagging activities as they work. In the current study, to achieve better scalability, we operationalized indicators of SMART operations and developed automated detectors using machine learning. We analyzed students' textual responses and interaction data collected from a mathematical learning platform where students are asked to thoroughly explain their solutions and are scaffolded in communicating their problem-solving process. Due to the rarity in data for one of the seven SRL indicators operationalized, we built six models to reflect students' use of four SMART operations. These models are found to be reliable and generalizable, with AUC ROCs ranging from .76-.89. When applied to the full test set, these detectors are relatively robust to algorithmic bias, performing well across different student populations and with no consistent bias against a specific group of students.

Keywords: self-regulated learning, SMART model, automated detectors

1. INTRODUCTION

Work over the last two decades has developed automated detectors of a range of behaviors and constructs in students' interaction with computer-based learning environments (Baker and Yacef, 2009). These detectors utilize log data collected from learning environments to infer the presence or absence of a complex behavior or a construct in student learning. For example, detectors have been built to identify student affect (e.g., Baker et al., 2004; Devolder et al., 2012; Labuhn et al., 2010), engagement (e.g., Baker et al., 2010; Paquette et al., 2014), and problem-solving strategies (Sao Pedro et al., 2013). Such detectors can be split into two broad categories: detectors for post-hoc analysis and detectors for real-time adaptation. The post-hoc analysis allows researchers to detect constructs retrospectively to understand their prevalence (e.g., Lee et al., 2011) and conduct further analysis (e.g., Botelho et al., 2018; Richey et al., 2021). Detectors designed to be run in real-time facilitate adaptive experiences and real-time feedback (Walonoski and Heffernan, 2006), as well as reports to teachers (Aguilar et al., 2021).

In particular, considerable work has been devoted to detecting and understanding behaviors and strategies involved in self-regulated learning (SRL). By examining student behavior patterns, automated detectors have been developed for a range of SRL-related constructs, including help avoidance (Aleven et al., 2006), gaming the system (Baker et al., 2004), setting goals (Azevedo et al., 2011; Biswas et al., 2010), and planning and tracking progress (Biswas et al., 2010). However, the specific constructs being modeled often have not been clearly linked to any of the growing number of theoretical models of SRL (although Farhana et al., 2021 and Hutt et al., 2021 are exceptions to this) and have mostly been operationalized in terms of high-level strategies that combine several behaviors treated as separate in SRL theories, rather than the finer-grained behaviors used in those theories (cf. Boekaerts, 1999; Bosch et al., 2021). Capturing fine-grained indicators of key aspects of SRL in terms of these theoretical models may yield a better understanding of the process of SRL and help EDM research make more direct theoretical contributions.

Self-regulation is a critical component of learning and has been positively associated with learning outcomes (Cleary and Chen, 2009; Nota et al., 2004; Zimmerman, 1990). In mathematics problem-solving, students who are skilled in SRL are able to effectively set goals, search for information, and direct their attention and cognitive resources to align their efforts with their objectives (Zimmerman, 2000). As a result, SRL facilitates successful problem-solving process (Cleary and Chen, 2009; Nota et al., 2004; Zimmerman, 1990) and enables students to acquire deep conceptual understanding (Labuhn et al., 2010). Given its benefits, theory-based interventions have been developed to promote SRL (Devolder et al., 2012). However, current SRL assessments, such as self-reports and think-aloud activities, are not sufficient to provide measurement at scale; at the same time, existing scalable SRL assessments based on automated detection in log data are typically not connected back to theory, making it difficult to use them in theory-driven interventions. SRL assessments based on automated detectors have therefore been used in more ad-hoc, system-specific interventions. These interventions have had mixed success, sometimes failing to impact learning outcomes or produce robust changes in student behavior (e.g., Roll et al., 2007). These measures have sometimes also correlated to learning outcomes in unexpected ways, often depending on specific details of the context of behavior that simpler automated detectors can fail to capture (Aleven et al., 2016; Ocumpaugh et al., 2021).

Therefore, in the current study, we develop automated detectors that identify fine-grained evidence of SRL constructs drawn from theory. This study does so in the context of CueThink, a digital learning application that focuses on enhancing middle school student mathematics problem-solving skills. Through the lens of the SMART model of SRL (Winne, 2017) (described in greater detail below), we identify and operationalize the following seven SRL indicators: 1) numerical representation, 2) contextual representation, 3) strategy orientation, 4) outcome orientation, 5) data transformation, 6) following plans, and 7) incorporating information. Due to the rarity of cases where students demonstrated strategy orientation, a detector could not be built. Therefore, six detectors were developed. We evaluate their performance and check them for algorithmic bias.

2. BACKGROUND

2.1. SRL AND THE SMART MODEL

Self-regulated learning (SRL) describes a series of self-generated thoughts, feelings, and behaviors that are systematically produced, orientating learners' attention and effort toward the attainment of goals (Zimmerman and Schunk, 2011). In the last three decades, several SRL models have been proposed to examine and explain the process and the enactment of SRL from socio-cognitive, motivational, and metacognitive perspectives (Panadero, 2017). For example, Zimmerman (2000) describes the process of SRL as three cyclical phases (forethought, performance, and self-reflection), in which learners analyze a task, execute the task, and assess the performance, respectively. In Pintrich's SRL model (2000), SRL is compounded by four phases (forethought, monitoring, control, and reaction and reflection), with each phase containing the regulation of cognition, motivation/affect, behavior, and context. Grounded in information processing theory, Winne and Hadwin (1998) characterize the process of SRL as four interdependent and recursive stages, in which learners: 1) define the task, 2) set goals and form plans, 3) enact the plans, and 4) reflect and adapt strategies when goals are not met. In our specific case, the design of the learning platform being studied (CueThink) resembles Winne and Hadwin's four-stage model. Therefore, we situate the current work in the four-stage model along with the SMART model (Winne, 2017) that was later developed to explain the cognitive operations involved in each stage.

Along with operations, COPES – a first-letter acronym of conditions, operations, products, evaluations, and standards – outlines five facets that are involved when learners accomplish the task in each stage of SRL (Winne, 1997). Specifically, within each stage, learners survey the conditions (C), elicit cognitive operations (O) to generate a product (P), and then evaluate (E) the product against a set of standards (S). Operations as a key facet in SRL describes the cognitive and metacognitive process of how learners interact with information, understanding the conditions, generating products, and evaluating products against standards.

In the first stage of SRL, learners define the task by creating a mental representation (Winne, 2004). By surveying the resources and constraints, learners outline what is known and what needs to be known. These resources and constraints can be internal to the learners, or external in the learning environment (Winne, 2017). For example, learners may have prior knowledge and/or misconceptions on a topic, and these are internal to the learners. On the other hand, the learners have access to external information, which can be found in the task itself or in the learning environment that the learners have access to (e.g., internet or peers). Both internal and external resources and constraints guide the development of problem representation, facilitating or impeding the student in accurately representing the problem (Winne, 2004). Once the problem is defined and represented, learners set goals and form plans. As stated in Winne (2017), the goals can be oriented toward the process of learning, focusing on the effort and the efficiency of the problem-solving process, or the goals can be set in terms of the product, emphasizing on the outcome. Goals may be further divided into subgoals, which allows learners to metacognitively monitor the progress of the work. When goals are identified, plans are typically formed to approach the goals. When the student enacts this plan, feedback may be generated, which may originate internally as a result of the learner monitoring the workflow and the progress, or externally as a response from a system, peers, or teachers (Winne, 2017). Learners can then acknowledge the feedback by updating goals to further the progress or correcting plans and strategies to accommodate any discrepancies.

The SMART model of SRL (Winne, 2017) was later proposed to further elucidate the cognitive processes involved. Specifically, the SMART model separates the "cognitive and behavioral actions applied to perform the task" into five categories: *searching*, *monitoring*, *assembling*, *rehearsing*, and *translating*. Each operation describes a way that learners cognitively engage and interact with information. For example, when working on a task, learners direct their attention to particular information (searching) and compare the information with a standard (monitoring), evaluating its relevance or importance. When relevant information is identified, students relate pieces of information to one another (assembling), in order to create a comprehensive understanding of the problem. When information does not fit into the current problem representation, learners manipulate how it is represented to find a solution (translating). Throughout the process, working memory is used to actively maintain and reinstate information (rehearsing).

These cognitive operations are an integral aspect of self-regulation: they help determine student success at completing each of the four SRL tasks, which, in turn, influences the progression of the problem-solving process (Winne, 2005). However, despite the SMART categories' importance for SRL, they are often difficult to observe or measure, as most learning activities (whether online or offline) do not fully reify the cognitive process involved in their learning tasks. Further, these operations may occur non-linearly, and multiple operations can be employed when completing the same task– making the measurement of these constructs challenging.

2.2. Challenges in SRL measurements

SRL has typically been measured using three approaches: self-reports, think-aloud activities, and log data collected in computer-based learning environments (Greene et al., 2013; Winne, 2010b). With traditional self-report studies, students are asked about their SRL process either outside of a task (i.e., before or after completing a task) or while working on a task. In decontextualized self-report (outside of a task), students report on the SRL strategies they plan to use or recall the strategies they used, using a pre- or post-task survey, respectively. The Learning and Strategies Study Inventory (LSSI) (Weinstein et al., 1987) and the Motivated Strategies for Learning Questionnaire (MSLQ) (Pintrich et al., 1991) are two self-report inventories that have frequently been adopted in studies. The instruments are administered outside of a task (e.g., Cho and Yoo, 2017; Roth et al., 2016)). Even though this approach is widely used, the nature of surveying cognitive processes outside of the task may lead to inaccuracies in the representation of cognition (Greene et al., 2013; Winne and Perry, 2000). For example, when recalling a cognitive process retrospectively, students may aggregate the out-of-context, self-reported experience across numerous tasks, failing to demonstrate the relationship between the task and the corresponding SRL strategies. For this reason, several studies have adopted in-context self-report (e.g., Sabourin et al., 2012; Taub et al., 2014; Winne et al., 2019) in which students are asked to tag their SRL strategies as they occur. However, in addition to disrupting the process of problem-solving, the surveys used for in-context tagging often only have a finite set of closed-ended responses, which makes it challenging to capture students' experience if it falls outside of the range and types of responses presented (Greene et al., 2013).

Other research has leveraged think-aloud activities that ask students to verbalize their cognitive processes when solving a problem (Greene et al., 2017). As with in-context self-reports, think-alouds give researchers an opportunity to identify processes that are contextualized in the problem-solving activity and are approximately concurrent with their occurrences. However, this process can suffer from an observation effect. Students being prompted to discuss their thinking process in real-time may alter that process and not provide an accurate representation of the processes they would engage in naturally (Bosch et al., 2021; Schooler et al., 1993). This, in turn, calls into question the validity of findings obtained using this type of measurement and whether they are generalizable to new students and contexts.

2.3. Use log data to measure SRL in computer-based learning environments

In addition to the issues identified above, self-report and think-aloud approaches are labor-intensive and time-consuming (Winne, 2010a), which make them difficult to scale. As such, a third approach, analyzing log data collected from computer-based learning environments, has emerged as a promising way to measure SRL.

Aleven and colleagues (2006) designed an exhaustive set of production rules to represent help-seeking behaviors within a geometry learning system and then compared these rules to student problem-solving steps to determine whether those steps were warranted by the current situation. Using a more bottom-up approach, Biswas et al. (2010) examined the sequences of student behaviors and modeled a range of SRL behaviors, including monitoring through explanation, self-assessment, tracking progress, and setting learning goals. Additionally, Segedy et al. (2015) utilized log data and coherence analysis to assess students' ability to seek out, interpret, and apply information in an open-ended learning environment, examining if a student's subsequent action is coherent based on the information presented.

Researchers have also used textual responses within dialogue-based learning systems to measure SRL. Graesser and colleagues (2007) used latent semantic analysis to study student conversations with animated pedagogical agents to assess and support SRL. Students who frequently use questions in a conversation can be interpreted as showing initiative, and engagement in monitoring can be inferred when students demonstrate in their responses that they feel they know the answer (Graesser and McNamara, 2010).

However, often the log data collected does not directly and straightforwardly relate to an SRL construct (Azevedo et al., 2017). Researchers must decide what data to use, what constructs to measure, and how to operationalize the constructs with the existing data (Kovanovic et al., 2016). SRL, as a process, covers a range of behaviors and strategies, so the constructs can vary depending on how SRL is conceptualized and also based on the design of the activity the learner is participating in. To ensure the validity of the operationalization, it is recommended that the operationalization should be conceptualized in terms of an SRL model and contextualized in the learning environment where the data is generated (Winne, 2010b).

2.4. Approaches to scaling up SRL measurements

In previous studies, machine learning (ML), knowledge engineering (KE), and several bottom-up approaches have been used to leverage log data to build automated detectors that measure SRL constructs. For example, San Pedro et al. (2013) leveraged ML to detect students' systematic inquiry behavior in a science learning platform. Using KE, Aleven and colleagues (2006) developed a rule-based model that detects effective and ineffective help-seeking behaviors. Bottom-up approaches that utilize sequence mining or cluster analysis have also been adopted in previous research that first examine student behaviors, interpreting the sequence or the commonalities of these behaviors in the context of problem-solving, and then identify and extrapolate behavioral patterns that resemble various SRL constructs (e.g., Biswas et al., 2010; Segedy et al., 2015).

When using machine learning, specifically supervised learning, detectors are built by training models that make binary classification predicting the presence or the absence of a behavior (Baker and Ocumpaugh, 2016). When training supervised ML models, two components are required: data that indicate the positive and negative examples of a behavior, referred to as the “ground truth”; and a set of features extracted from these examples. A machine learning algorithm will then be applied to learn the differences in the features between the positive and negative examples and find the relationships that are indicative of the behavior. With the relationships learned, the model makes predictions based on the features provided.

As the process mainly relies on data and algorithms, the time spent in developing ML models focuses more on the process of labeling ground truth, feature engineering, and choosing the right algorithms. The algorithms identify the relationships between features of the data and labels, saving the researcher the manual work of identifying the relationships between observable behaviors and targeted constructs, which is emphasized in knowledge engineering approaches. This automation means that ML models have the potential of finding underlying relationships that are less explicit or may be overlooked by domain experts (see discussion in Paquette and Baker, 2019), and may also be less susceptible to human biases when encoding the relationships. However, machine learning can be impacted by human biases in the training labels. For example, Okur et al., (2018) shows that using video coding to label affect can result in culturally biased data.

Although building ML models does not require extensive manual work at the model-building stage, a substantial amount of data is needed for the algorithms to learn from, in order for the models to be successful. Considerable effort often also goes into feature engineering when working with learning system data (Paquette et al., 2014). Additionally, despite the potential accuracy of ML models, they are generally hard to interpret (Paquette and Baker, 2019). ML models often do not provide a straightforward explanation of how predictions are made, which can impede the translations between model predictions and theoretical insights. This problem is worsening with the increasing adoption of highly-complex neural network algorithms (Webb et al., 2021). This difficulty in explaining the model or its predictions makes this approach hard to adopt in situations where interpretability is prioritized. Understanding what factors contribute to the presence of a learning behavior could provide valuable insight when designing learning environments or interventions that support learning. With the recent emphasis on building explainable and interpretable models (e.g., Conati et al., 2018; Mu et al., 2020; Webb et al., 2021), methods such as SHapley Additive exPlanations (SHAP) (Lundberg et al., 2019) value, have been developed to interpret ML models. However, a critical issue is raised in Swamy et al. (2022), in which they found that the results derived from various explainers (methods used to explain predictive models) do not agree on feature importance. Therefore, they suggest a careful evaluation when using the explainers and call for further examination of the topic. Nonetheless, the recent trend of developing and improving methods that can be used to interpret ML models, makes ML a promising approach in building automated detectors that can be better inspected by developers and stakeholders.

On the other hand, detectors created using the KE approach rely on domain experts who develop a set of rules that capture the knowledge required to identify the behavior (Muldner et al., 2011; Paquette and Baker, 2019). A KE model of a behavior is based on existing literature (if it exists) and generally accepted definitions of that behavior, as well as on domain experts who identify and interpret action patterns that are its indicators.

As the knowledge engineering model is manually developed and directly drawn from the expert's knowledge of the specified behavior (Paquette et al., 2014), a major advantage of this method is that it doesn't require large amounts of labeled data of the behavior, unlike ML models. The process of knowledge engineering also offers a more transparent and interpretable model that gives insights into how a model is developed, what the action patterns comprising the model mean, and how learner behaviors are associated with the model (Paquette and Baker 2019). KE models also can be more generalizable than machine-learned models (as seen in Paquette et al., 2014) which can be attributed to the experts' abilities to capture deeper underlying features of the behavior, whereas an ML model may capture correlations that are limited to the current dataset in use. However, KE models run a risk of encoding the biases of domain experts as they develop the detectors. Additionally, KE models are sensitive to even slight differences in the rules, as it can lead to significant differences in the inferences drawn from the model (Kovanovic et al., 2016). In addition, some behaviors are difficult for a human knowledge engineer to precisely characterize. For instance, when a behavior cannot be easily described and distilled into action patterns by the researchers, it may be better to use ML models. A specific case of this is when an exact numerical threshold has to be selected, or implicit correlations between behaviors and target constructs (not even known to the human expert themselves) have to be identified.

In addition to ML and KE, Biswas et al. (2017) outlines several other approaches, such as sequence mining and cluster analysis, that have been used in previous studies to discover and interpret behavioral patterns that reflect the use of self-regulation (e.g., Biswas et al., 2010; Dever et al., 2022; Kinnebrew et al., 2013; Segedy et al., 2015). These bottom-up approaches first examine student behaviors over time or at a given circumstance, interpreting the sequence and/or the commonalities of these behaviors in the context of problem-solving. For instance, by leveraging a novel sequential mining technique, analyzing how students interact with an interactive virtual agent, Kinnebrew and colleges (2013) discovered that low- and high-performing students produce different sequences of actions during learning, and that some of these patterns can be meaningfully linked to SRL.

As opposed to KE, in which domain experts develop a set of production rules that capture the knowledge required to identify a self-regulated behavior, bottom-up methods focus more on discovering and identifying these behaviors, interpreting them, situating them in the problem-solving process, and relating them to the use or the lack of use of SRL strategies. However, because of the bottom-up nature of these approaches, the patterns identified using these methods sometimes may not be interesting or informative to investigate, or these approaches may identify behaviors that are already known (Biswas et al., 2017). For example, Kock and Paramythis

(2011) used cluster analysis to analyze the sequence of help usage over time, but primarily discovered patterns that resembled the behaviors already documented in Aleven et al. (2006).

2.5. ALGORITHMIC BIAS

In order to use detectors at scale, we must ensure that they will be valid for the entire populations they are scaled to rather than only subgroups of students. Recent evidence suggests that many published detectors are prone to algorithmic bias, functioning better for some populations of learners than others (Baker and Hawn, 2022). However, there has been limited attention to algorithmic bias within the field of educational data mining, where analyses of algorithmic bias are rare and even overall population demographics are only reported in 15% of publications (Paquette et al., 2020). Verifying algorithmic bias is important any time when detectors will be deployed onto a learning platform and used to support a diverse population of learners. It would be highly problematic for many reasons to deploy a detector that functions significantly less well for some learners than others. Thus, it is important to evaluate detector effectiveness across demographic groups before deploying and using the detectors at scale.

As defined in Kizilcec and Lee (2020), algorithmic bias describes the problem where a data-driven predictive model functions better for some populations than others, producing disparate and poorer impact for historically underrepresented or protected groups. As the predictions are often used to inform decisions and actions, algorithmic bias in a model can cause unfairness in the allocation of resources and misplacement of treatment (Kizilcec and Lee, 2020). To understand the cause of the issue, Kizilcec and Lee (2020) examined the development of algorithmic systems and outlined a number of factors that can potentially contribute to the issue, categorizing them in the three phases of algorithmic development – measurement, model learning, and action.

In the process of measurement, selecting a suitable target variable and collecting representative data are two rudiments in developing a fair algorithmic system. More specifically, careful consideration is needed when choosing the target variable in predictive modeling, as target variables may encode biases inherited from historical patterns, prejudice, or discrimination (Karumbaiah and Brooks, 2021). Additionally, when building a predictive model, the training data should be representative for whom the model is built for. For instance, Ocumpaugh et al., (2014) show that affect detectors' performance degrades if they are applied to student groups whose data were not represented in the training set. Specifically, they found that the detectors demonstrated better accuracy for students from rural, suburban, and urban regions if the detectors rely on data drawn primarily from the same demographic grouping. Similarly, differences in model reliability are also found in Gardner et al., (2019), which evaluated the performance of a model that predicted students' dropout rate based on gender. Models trained in courses with student populations where males were better represented than females were less successful at predicting outcomes for female students. These findings highlight the need for a close examination of the data and measurements, and careful consideration of the generalizability and the validity of models to ensure their fairness.

One avenue to evaluate the fairness of predictive methods is through slicing analysis. Slicing analysis evaluates a predictive model's performance by slicing the results of that model across different dimensions or categories in the test set (Sculley et al., 2018). As such, slicing analysis provides a more granular examination of a predictive model and makes it possible to evaluate the relative performance or fairness of the model across subgroups.

3. Current Study

In the current study, using machine learning, we build automated detectors of SRL constructs from a theory-driven lens. Using a dataset of 79 students as they interacted with the online math learning platform CueThink, we first examine the learning environment, understanding how students interact with the platform and the context of how the log data is generated. Based on the context and the log data available, we identify relevant theoretical constructs grounded in the SMART model. In particular, the following seven SRL indicators relating to four cognitive operations in the SMART model are identified for investigation: 1) numerical representation, 2) contextual representation, 3) strategy orientation, 4) outcome orientation, 5) data transformation, 6) following plans, and 7) incorporating information.

When developing the detectors, we use text replay to code student interactions for each indicator. These labels are then used as ground truth for machine learning. We distill a variety of features from the log data to represent multiple aspects of a student's interaction, including the number of responses and the content in the responses. The ground truth and the features are then input into a machine learning process, training a model to emulate human coders' judgment and making predictions on the presence or absence of an SRL indicator. We demonstrate that trained detectors provide accurate detection, suitable for real-time use. Then, through slicing analysis, we evaluate the performance of our models across different demographic groups and demonstrate that the detectors are fair overall and are not consistently biased against any specific student groups.

4. Data

4.1. Learning Environment

CueThink is a digital learning application that focuses on enhancing middle school student math problem-solving skills by encouraging students to engage in self-regulated learning and develop math language to communicate problem-solving processes. CueThink asks students both to solve a math problem and to create a shareable screencast video that provides the student's answer and also demonstrates their problem-solving process. As Figure 1 shows, CueThink structures a problem into a *Thinklet*, a process that includes four phases—*Understand, Plan, Solve,* and *Review*—that closely align with Winne and Hadwin's model of SRL (1998).

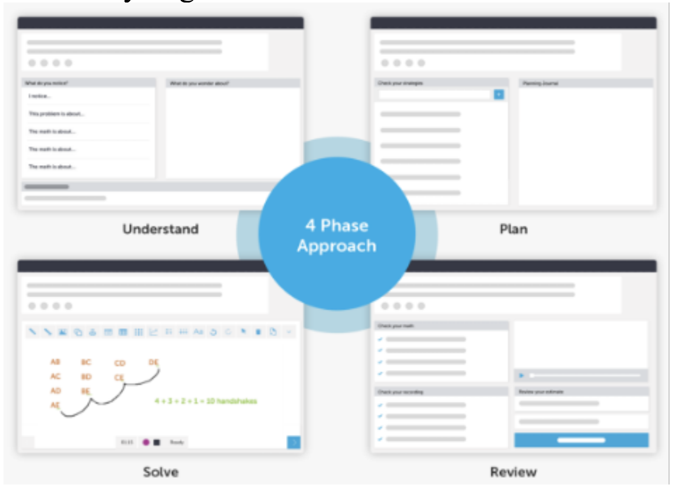


Figure 1: Screenshots of CueThink's 4 Phase Approach.

Each phase of the *Thinklet* (outlined in Table 1 and described in more detail below), asks students to focus on a different part of the problem-solving process. While working on a *Thinklet*, students can move freely across the four phases, including going back to a previous phase or skipping phases.

Starting with the *Understand phase*, students read a problem and provide text-based responses to three questions: (1) "What do you notice?" (2) "What do you wonder?" and (3) "What is your estimated answer to the problem?" This phase encourages students to actively look for information in the problem and create a representation of the problem space. Thus, students demonstrate their understanding of what they know and what they need to know at this phase.

In the *Plan* phase, students build on what they have established in the *Understand* phase by planning how they will solve the problem. Students are first prompted to select what strategies they will use to solve the problem. They may choose from a predefined strategy list (i.e., draw a picture, model with an equation, work backwards, etc.) or define their own strategies. Once the student has selected which strategies they will use, the student is prompted to write a plan on how they will use the strategies to solve the problem.

In the *Solve* phase, students explain and present their answers. Specifically, they create a screencast video using an interface that provides them with a whiteboard and mathematical tools (i.e., number lines, ruler, etc.).

In the *Review* phase, students provide the final answer to the math problem and reflect on whether the answer makes sense and whether their communication is clear, using checklists to scaffold their reflection.

Once students have completed the problem, they share their screencast explanation for *Peer Review*. In this phase, teachers and peers annotate both the textual responses and video, often asking the student for their underlying reasoning or why the student picked specific methods. These annotations are then sent back to the video's author for possible revision.

Table 1: Summary of Tasks by *Thinklet* Phase.

Phase	**Tasks Description & Data Types**
Understand	What I notice is (*textual response*) What I wonder is (*textual response*) Estimate your answer (*textual response*)
Plan	Choose your strategies (*select all that apply, textual response*) Planning journal (*textual response*)
Solve*	Video creation tools (*Whiteboard, math tools, and recording tools)*
Review	Check your math (*select all that apply*) Check your recoding (*select all that apply*) Review your estimate (*textual response*) Final answer (*textual response*)

* Student activity in the *Solve* phase is not used in this paper's analyses

4.2. STUDENTS DEMOGRAPHICS

In this study, 79 students in grades 6 and 7 at a suburban school in the southwestern U.S. used CueThink during the 2020-21 school year. The school contains a diverse student population, with around 40% Hispanic or Latino, 40% White, 15% African American, and 5% Asian students. Students' self-reported demographic information on gender and race/ethnicity was collected. For gender, students could choose to identify as male, female, non-binary, or leave the question blank. For race/ethnicity, options included African American, Hispanic/Latinx, White, Asian, Native American, two or more races, other, or prefer not to say. Students reporting "other" for their race/ethnicity were provided the option to give detail.

4.3. LOG DATA

CueThink was used in six classrooms over multiple weeks, with teachers assigning problems for students to complete in the application. We collected log files that reflect how students use the application and their problem-solving process. On average, students spent 5.2 hours in CueThink and 1.8 hours working on each *Thinklet,* with each *Thinklet* containing one math problem. For each *Thinklet*, we collected the questions students answered and their textual responses at each phase. In this study, we analyzed textual and click-stream data but did not analyze data from the videos. In total, we collected 349 *Thinklets* from 79 students working on 24 different problems. Of those 349 *Thinklets*, not all were first attempts. Students have the opportunity to revise their work, which creates another *Thinklet*. In those cases, it is possible that a student would not have gone through the entire problem-solving process. Of the total number of *Thinklets*, 146 were duplicate attempts.

5. BUILDING AUTOMATED DETECTORS

In this section, we described a multi-step process of using machine learning to build automated detectors of self-regulated behaviors. When building these detectors, we first distilled human-readable text replays from log data (Baker et al., 2006). Using these text replays, we identified and operationalized qualitative categories that corresponded with SRL constructs, grounding the operationalization in Winne's SMART model. We then labeled the self-regulated behaviors, generating ground truth data. Feature engineering and feature distillation were then conducted. Using the ground truth data and features distilled, we trained models using XGBoost to predict the presence or absence of the SRL constructs[1].

5.1. TEXT REPLAYS OF INTERACTION LOGS

To facilitate the inspection and exploration of the data, we used text replays (Baker et al., 2006). This method presents segments of interaction data (referred to as clips) in a human-readable presentation. This process facilitates both the initial exploration of the data (such as in section 5.2) along with the final coding process (section 5.3). Clips are then viewed by human coders who label them accordingly (Baker et al., 2006). Previous studies have used text replay coding to label student affect, disengagement, and learning strategies, such as gaming the system (Baker and Carvalho, 2008), confrustion (Lee et al., 2011), player goals (DiCerbo and Kidwai, 2013),

[1]XGBoost Models: https://github.com/JZ2655/EFMATH_JEDM2022.git

and SRL strategies such as whether a student is using a table to plan their analyses (Sao Pedro et al., 2013). This approach achieves a level of reliability similar to classroom observations and is 2-6 times faster compared to other methods of generating labels, such as classroom observations, screen replay, and retrospective think/emote-aloud protocols (Baker et al., 2006).

The length and the grain-size of text replay clips can vary depending on the granularity of the predictions the researcher intends to make. Because this study seeks to detect cognitive SRL operations in the problem-solving process, which requires a comprehensive examination across questions and phases, the log files were delineated into clips on the level of entire *Thinklets*. Each clip contains a student's actions and text-based responses that were submitted as that student worked through the four phases to produce a single *Thinklet*. The clips were distilled from log files and presented using a Python window, shown in Figure 2. As the indicators operationalized in the current study mainly reflect SRL constructs that are likely to be observed in the first two stages in the problem-solving process, where learners define tasks and form plans, video data from the *Solve* phase was not converted and was not included in the text replays.

5.2. CONSTRUCT OPERATIONALIZATIONS

To identify constructs to detect, we first examined the clips containing student responses in *Thinklets* and coded student responses for indicators of SRL – qualitative categories that correspond with SRL constructs (we discuss the details of exactly how the data was coded in section 5.3). The definitions of the indicators we coded were developed through dialogue between the research team and system developers. This process followed the recursive, iterative process used in (Weston et al., 2001) that includes seven stages: conceptualization of codes, generation of codes, refinement of the first coding system, generation of the first codebook, continued revision and feedback, coding implementation, and continued revision of the codes (Weston et al., 2001). The conceptualization of codes included a review of related literature, including several theoretical frameworks and perspectives (Bandura, 1986; Boekaerts, 1999; Efklides, 2011), primarily focusing on the SMART model (Winne, 2017). Using grounded theory (Charmaz 1983), we identified common behaviors that were (1) indicative of SRL as characterized by Winne's SMART model (Winne, 2017) and (2) salient in the log files. A draft lexicon and multiple criteria were generated for a coding system to help identify these constructs.

Given the learning environment's design and the available data, our efforts focused on defining behaviors related to four categories of cognitive operations (namely, the monitoring, assembling, rehearsing, and translating operations from the SMART model) as they are frequently employed in the initial stages of SRL as learners define tasks, set goals, and make plans. Following the process used in (Weston et al., 2001), two coders coded a set of clips together, identified seven SRL indicators, and outlined the criteria for each indicator, and created a rubric. These indicators are numerical representation, contextual representation, strategy orientation, outcome orientation, data transformation, following plans, and incorporating information. The draft coding manual was discussed with all members of the research team and developers and designers at CueThink to build a common understanding of the criteria and constructs being examined as well as the features of the system to gain feedback for further refinement. This process was repeated until the entire team had reached a shared understanding of the criteria and constructs being examined by the codebook. The SRL indicators identified, the criteria, and alignment with the SMART model are included in Table 2.

Numerical and contextual representation consider a learner's process of creating a problem representation, which often occurs in the initial stage in the problem-solving process (i.e., define the task), outlined in the four-phase model of SRL (Winne and Hadwin, 1998). In problem representation, learners create a problem space by identifying information they know and information they need to know. These two SRL indicators encode how learners represent and process information in math problems, denoting if numerical components and/or contextual details are noted. We consider both of these processes to reflect assembling in the SMART model as students are creating their representation of the data from the information provided. There may also be overlap with translating in some cases, especially if the question provides a different representation to the one the students use. However, as this is not always the case, we primarily consider both indicators to reflect assembling actions and tag translating actions in a different code (see below).

Strategy and outcome orientation also reflect student assembling behaviors. Both of these indicators concern how students set their goals and form plans for the problem-solving process. These two indicators demonstrate a difference in focus (process vs. output). However, these two indicators do not have to be mutually exclusive. A comprehensive goal may contain both a process component that reflects the strategies the learner plans to adopt to solve the problem as well as an estimation of what the learner believes the outcome should be.

As mentioned above, students may change how information is provided when forming a representation. *Data transformation* reflects behaviors that are associated with the translating operation. In data transformation, the learner manipulates the ways information is represented to them in the problem to find a solution.

Following plans as an indicator reflects scenarios where learners incorporate previously selected strategies into their plans, documented in the planning journal task in the learning platform. This alignment between the strategies selected and strategies incorporated in the plans demonstrates learners' enactment of the monitoring operation, showing that learners are actively checking and monitoring the congruence of their responses.

Lastly, also observed in the planning phase, *incorporating information* reflects behaviors where learners correctly and meaningfully incorporate previously assembled information into their plans. These behaviors reflect an engagement with the rehearsing operation, in which the learners review information that has been previously noted.

Table 2: SRL Indicators Coded through Text Replays.

SMART Category	**SRL Indicator**	**Working Definition**
Assembling	Numerical Representation (NR)	The learner's representation of the problems includes numerical components and demonstrates a level of understanding of how the numerical values are used in the math problem.
Assembling	Contextual Representation (CR)	The learner's representation of the problem includes contextual details relating to the setting/characters/situations within the given math problem.

Assembling	Strategy Orientation (SO)	Learners explicitly state a plan for how they will find the answer for the given math problem, decomposing information into a step-by-step process.
Assembling	Outcome Orientation (OO)	The learner provides only a numerical estimate of the final answer for the given math problem, suggesting that learners are focused on the output instead of the process itself.
Translating	Data Transformation (DT)	The learner manipulates the ways information is represented to them in the problem to find a solution. This suggests active problem-solving.
Monitoring	Following Plans (FP)	When making a plan, learners correctly and meaningfully incorporate the strategies they selected into the plan.
Rehearsing	Incorporating Information (II)	When making a plan, learners correctly and meaningfully incorporate information previously assembled into the plan.

5.3. Coding the Data

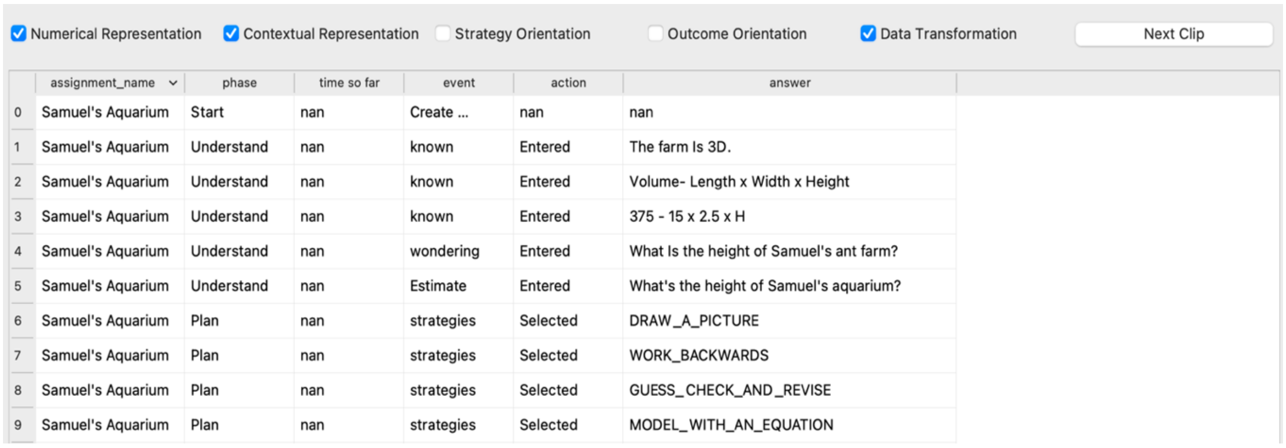
After constructs were operationalized and defined, we proceeded to code the remainder of the data. Two coders (the same as in the previous section) completed the text replay coding in three phases: preliminary coding (discussed above), separate coding (two coders per clip; for establishing inter-rater reliability), and individual coding (one coder per clip; to code the full set of clips used in the detectors).

Table 3: Inter-Rater Reliability in Separate Coding.

SRL Indicator	**IRR Kappa**
Numerical Representation (NR)	0.83
Contextual Representation (CR)	0.63
Strategy Orientation (SO)	0.74
Outcome Orientation (OO)	0.78
Data Transformation (DT)	0.74
Following Plans (FP)	0.75
Incorporating Information (II)	0.80

The two coders each used the codebook/rubric to code the same set of clips separately. They then compared the labels and computed the inter-rater reliability (IRR) kappa. For constructs with low kappa, the two coders discussed their differences in labeling and conducted another round of coding. This step of separate coding and comparing is repeated until acceptable reliability is established. After two rounds of coding, the two coders reached an acceptable IRR above 0.60 for all five SRL indicators (M=0.75), shown in Table 3.

Once reliability was established, the coders moved on to the individual coding where they split the rest of the clips and coded them individually. Each construct was considered over the entire *Thinklet*. Thus, in total, the two coders coded 349 clips. However, in order to consistently examine the entire problem-solving process, 167 clips that were marked incomplete because students stopped before completing the entire problem were excluded. Of the remaining 182 clips, coding resulted in the following distribution of labels: 64% numerical representation, 77% contextual representation, 8% strategy orientation, 72% outcome orientation, 73% data transformation, 47% following plans, and 41% incorporating information. These were produced by 72 students, who, on average, each contributed 3 clips (max=4, min=1, median=3).


☑ Numerical Representation ☑ Contextual Representation ☐ Strategy Orientation ☐ Outcome Orientation ☑ Data Transformation

	assignment_name	phase	time so far	event	action	answer
0	Samuel's Aquarium	Start	nan	Create ...	nan	nan
1	Samuel's Aquarium	Understand	nan	known	Entered	The farm Is 3D.
2	Samuel's Aquarium	Understand	nan	known	Entered	Volume- Length x Width x Height
3	Samuel's Aquarium	Understand	nan	known	Entered	375 - 15 x 2.5 x H
4	Samuel's Aquarium	Understand	nan	wondering	Entered	What Is the height of Samuel's ant farm?
5	Samuel's Aquarium	Understand	nan	Estimate	Entered	What's the height of Samuel's aquarium?
6	Samuel's Aquarium	Plan	nan	strategies	Selected	DRAW_A_PICTURE
7	Samuel's Aquarium	Plan	nan	strategies	Selected	WORK_BACKWARDS
8	Samuel's Aquarium	Plan	nan	strategies	Selected	GUESS_CHECK_AND_REVISE
9	Samuel's Aquarium	Plan	nan	strategies	Selected	MODEL_WITH_AN_EQUATION

Figure 2: Screenshot of Text Replay Coding Window.

5.4. Feature distillation

After data coding yielded a set of labels, two sets of features were distilled from the clips that had been coded, to be input into the algorithms used to build the detectors. Both sets of features consist solely of features that can be extracted and used in real-time. As video data from the *Solve* phase were not included in text replays, we did not extract data from this phase in feature distillation.

The first set of features (N = 10) were distilled at the *Thinklet* level. These features were designed to provide an overview of the completeness of a *Thinklet* by examining the number of responses (as opposed to the content in the responses) in a *Thinklet*. For example, we distilled the number of questions students answered in a *Thinklet*, a binary indication showing if each prompt or question (e.g., "what I notice") is answered, and the number of responses (text entries) in each phase. Additionally, to understand the strategies that students selected in the *Plan* phase, we also created a feature that counts the number of strategies a student selects among the top two strategies used by peers for the same problem.

The second set of features was designed to examine the content and the linguistic features of students' text-based responses. These features (N = 90) were first extracted at the response level and then aggregated to the phase level. These aggregations were calculated for the *Understand*, *Plan*, and *Review* phases.

Specifically, we distilled whether each response: 1) contains a numerical value, 2) consists of only numerical values, 3) has mathematical operation signs, 4) contains a question (if it contains a question mark or uses keywords such as "wonder", "why", etc.), 5) uses language that indicates the formation of a plan (e.g., the use of keywords like "plan", "I will", "going to", etc.) , and 6) is the exact repetition of a previous answer. These criteria generate a set of binary variables for each response. We averaged these binary variables across the responses within a phase, creating 18 features for each *Thinklet*.

Additionally, in each response, we counted the number of 7) characters, 8) words, 9) numerical values, 10) verbs, 11) nouns, and 12) pronouns. Features 10-12 were counted using Udpipe, a natural language processing toolkit (Wijffels et al., 2017). We also 13) counted the number of keywords used from a predefined list that provides the context of each problem; and 14) computed how similar each response is to the problem item using the Smith-Waterman algorithm (Smith and Waterman 1981). For these continuous variables, we computed the mean, standard deviation, and max of the values for each phase, creating 72 features.

Features distilled from the two sets were combined. In total, 100 features were extracted from each *Thinklet* and were then used to construct the automated detectors. Note that we did not extract any features from the video that students make in the current work. Similarly, we did not use any of the audio from the video (or transcription thereof) for any features.

5.5. MACHINE LEARNING ALGORITHMS

We used the scikit-learn library (Pedregosa et al., 2011) to implement commonly-used classical machine learning algorithms, including Logistic Regression, Lasso, Decision Tree, and Random Forest, and used the XGBoost library (Chen and Guestrin, 2016) implementation of Extreme Gradient Boosting (XGBoost). XGBoost outperformed other algorithms in all cases; we therefore only discuss the XGBoost results below.

XGBoost uses an ensemble technique that trains an initial, weak decision tree and calculates its prediction errors. It then iteratively trains subsequent decision trees to predict the error of the previous decision tree, with the final prediction representing the sum of the predictions of all the trees in the set. We tested the detectors with 10-fold student-level cross-validation, to verify generalizability to new students. For this approach, the dataset was split into 10 student-level folds, meaning that in cases where students had multiple *Thinklets,* all of their data would be contained within the same fold and at no time could data from a student be included in both the training and testing set. Nine folds were used to train the model, and the trained model was used to make predictions for the 10th fold. Each fold was used as the test set once.

Models were evaluated using the area under the Receiver Operating Characteristic curve (AUC ROC), which indicates the probability that the model can correctly distinguish between an example of each class. An AUC ROC of 0.5 represents chance classification, while an AUC ROC of 1 represents perfect classification. Results were calculated for each fold and averaged to yield one AUC ROC score per detector.

6. RESULTS

In this section, we report the results of the models developed in the previous section. Specifically, we evaluated the ROC AUC of the models using 10-fold student-level cross-validation, examined the feature importance, understanding how features contribute to the performance of a model, and checked for algorithmic bias.

6.1. MODEL PERFORMANCE

Due to the rarity of strategy orientation (only 14 clips were labeled with this construct), a detector could not be built for this construct. Automated detectors were built for the other six constructs. As shown in Table 4, the average AUC ROC derived from 10-fold student-level cross-validation for the six detectors ranges from 0.76 to 0.89 with a standard deviation across folds around 0.1 for all six constructs. In specific, the average AUC ROC is 0.894 for numerical representation (NR), 0.813 for contextual representation (CR), 0.761 for outcome orientation (OO), 0.815 for data transformation (DT), 0.808 for following plans (FP), and 0.803 for incorporating information (II). These findings suggest that the detectors were generally successful at capturing these six SRL constructs. We also calculated the standard deviations (SD) of the AUC ROCs across the 10 folds for each detector to investigate the variability across folds.

Table 4: Detector Performance Measured by AUC ROC.

SRL Indicator	**AUC ROC (SD)**
Numerical Representation	0.894 (.078)
Contextual Representation	0.813 (.132)
Outcome Orientation	0.761 (.076)
Data Transformation	0.815 (.163)
Following Plans	0.808 (.099)
Incorporating Information	0.803 (.103)

6.2. FEATURE IMPORTANCE

To better understand the detectors as well as to inform our understanding of how these features relate to the constructs, we calculated the SHapley Additive exPlanations (SHAP) (Lundberg et al., 2019) value of each feature. Recent work has found that different explainability methods can produce very different estimates of feature importance (Swamy et al., 2022), making it essential to choose an appropriate method of assessing feature importance. SHAP has several advantages for the current project, given its solid theoretical foundation, its property of being model-agnostic, its ability to produce consistent and locally accurate attribution values, and its better alignment with human intuition than other approaches (Lundberg et al., 2019).

In the current study, SHAP value was computed for each feature within each test set. These values were then averaged across the 10 test sets and ranked based on their absolute values. Of the 100 features used, Table 5 reports the top five features with the highest absolute SHAP values for each detector. To understand the directionality, we examined the average SHAP values of the features listed. In the last column in Table 5, the positive and negative signs are used to reflect the average SHAP value, denoting the directionality of a feature. A positive average value indicates that the feature is a positive predictor of the SRL indicators, suggesting that the higher the value, the more likely the model is to infer the presence of an SRL indicator. Features denoted with a negative sign indicate a negative average SHAP value, as these features positively predict the absence of the SRL indicators.

Table 5: The Top Five Features from each Detector and their Directionality.

Feature Phase	Feature	+/-
Numerical Representation		
Understand	Mean N of responses that give numerical values	+
Understand	Max value of the similarity feature which indicates how parallel a student's response is to the original problem	+
Understand	SD of the similarity feature	+
Understand	Total N of responses	+
Plan	Avg value of the similarity feature	+
Contextual Representation		
Understand	N of responses to the "what do you notice" question?	+
Understand	Avg N of keywords used	+
Understand	SD of the N of characters used	-
Thinklet	Total N of responses	+
Plan	Max value of the N of characters used	+
Outcome Orientation		
Review	Avg N of keywords used	+
Review	Avg N of words used	+
Understand	SD of the N of numerical values used	+
Understand	Is there a response to the "what is your estimated answer" question?	+
Review	Avg N of nouns used	+
Data Transformation		
Plan	N of strategies selected that were among the most common strategies used by peers	+
Understand	SD of the similarity feature	+
Plan	SD of the N of characters used	+
Understand	SD of the N of nouns used	+
Plan	Max value of N of words used	+

Following Plans		
Plan	SD of the N of verbs used	-
Plan	SD of the N of pronouns used	-
Review	Avg N of pronouns used	+
Plan	Avg N of verbs used	+
Review	Avg N of verbs used	+
Incorporating Information		
Plan	Avg N of characters used	-
Plan	Avg N of pronouns used	+
Plan	Max value of N pronouns used	-
Plan	Max value of N words used	-
Plan	SD of N of nouns used	-

We note that of the 30 features listed in Table 5, 11 are from the *Understand* phase, 13 are from the *Plan* phase, and 5 are from the *Review* phase. In other words, behaviors in the early phases contributed more heavily to the predictions. This finding aligns with how the *Thinklets* were initially coded. Specifically, the coders primarily examined student responses in the *Understand* phase for numerical and contextual representation as this phase contains information demonstrating how students assemble information and create a problem representation; the coders examined the *Thinklet* more broadly when coding for other SRL indicators, as they represent behaviors that span across phases. One can also notice how various aspects of a linguistic feature are used in predicting the indicators. As the number, average, and max values of a linguistic feature (e.g., words, nouns, or verbs) indicate the frequency of its presence, standard deviation reflects the consistency of use of the linguistic feature across responses within a phase. These differences in what linguistic features are used and how they are used reflect differences in learners' responses in a *Thinklet*, representing various behaviors and attention when eliciting different SRL strategies. The implications of specific features will be discussed in detail in the discussion section.

To better understand the distribution of these top features in the dataset, we report the mean and frequency of the features across *Thinklets,* in the Appendix. We present the mean for features with continuous values (e.g., the total number of responses in the *Understand* phase) to give a sense of a typical value for the feature, and calculate the percentages for features with binary values (e.g., is there a response to the "what is your estimated answer" question) to reflect the prevalence of the feature.

6.3. ALGORITHMIC BIAS

Algorithmic bias describes the problem where a data-driven predictive model functions better for some populations than others, producing disparate and poorer impact for historically underrepresented or protected groups (Kizilcec and Lee, 2020). To validate our detectors, we tested the model performance in different student populations based on gender and race/ethnicity using slicing analysis (Gardner et al., 2019). Specifically, utilizing the predictions made in the testing sets, AUC was computed for each subgroup of students in the data for which we received data on group membership. However, due to sample size, comparisons were not possible for gender non-binary students (N=2), Asian students (N=2), or Native American students (N=0).

As Table 6 shows, the difference in model performance measured by AUC between male and female students is small to moderate, ranging from 0.01-0.11 for the six detectors. However, though some differences reached 0.11, the directionality was not consistent between male and female students. Furthermore, all detectors achieved AUC ROC over 0.7 for both male and female students. The detectors for numerical representation, contextual representation, and incorporating information performed somewhat better for female students (AUC_{NR} = .93, AUC_{CR} = .75, AUC_{II} = .84) than for male students (AUC_{NR} = .82, AUC_{CR} = .74, AUC_{II} = .72), while detectors for outcome orientation, data transformation, and following plans performed somewhat better for male students (AUC_{OO} = .78, AUC_{DT} = .88, AUC_{FP} = .82) than for female students (AUC_{OO} = .74, AUC_{DT} = .87, AUC_{FP} = .76).

Table 6 also shows the analysis of algorithmic bias in terms of race/ethnicity, comparing the AUC between student racial/ethnic subgroups that had more than 5 students in our sample: African American, Hispanic/Latinx, and White. Small to moderate differences were observed across the three groups, though the differences were not consistent (i.e., no racial/ethnic group consistently had the best-performing detectors). Furthermore, performance remained over 0.7 for all six detectors across all groups, with only two exceptions (discussed below). When detecting numerical representation and contextual representations, the detectors performed somewhat better for White students (AUC_{NR} = 0.96, AUC_{CR} = 0.80), than for African American (AUC_{NR} = 0.92, AUC_{CR} = 0.75) and Hispanic/Latinx (AUC_{NR} = 0.88, AUC_{CR} = 0.72) students. However, the outcome orientation detector had somewhat higher performance for Hispanic/Latinx students (AUC_{OO} = 0.81), than for White (AUC_{OO} = 0.80) and African American (AUC_{OO} = 0.71) students. The data transformation detector performed better for African American students (AUC_{DT} = 0.92) than for Hispanic/Latinx (AUC_{DT} = 0.91) and White (AUC_{DT} = 0.83) students. The detectors perform similarly across African American (AUC_{FP} = 0.75, AUC_{II} = 0.85), Hispanic/Latinx (AUC_{FP} = 0.79, AUC_{II} = 0.71), and White students (AUC_{FP} = 0.79, AUC_{II} = 0.78) when detecting the behaviors of following plans and incorporating information.

Table 6: Detector Performance by Gender and Racial/Ethnic Groups.

			Gender			Race/Ethnicity					
	All Students (k-fold)	All Students (Pooled)	Male	Female	Left Blank	African American	Hispanic/ Latinx	White	Prefer Not to Say	Other	Two or more races
N. students	72	72	33	28	9	6	18	8	19	14	5
N. clips	182	182	81	73	24	20	38	19	50	37	12
NR	0.89	0.89	0.82	0.93	0.97	0.92	0.88	0.96	0.86	0.85	0.86
CR	0.81	0.80	0.74	0.75	0.94	0.75	0.72	0.80	0.90	0.65	0.78
OO	0.78	0.75	0.78	0.74	0.72	0.71	0.81	0.80	0.74	0.78	0.46
DT	0.82	0.86	0.88	0.87	0.78	0.92	0.91	0.83	0.84	0.82	0.86
FP	0.81	0.80	0.82	0.76	0.83	0.75	0.79	0.79	0.91	0.73	0.83
II	0.80	0.80	0.74	0.84	0.77	0.85	0.71	0.78	0.78	0.86	0.91

Two exceptions were found to this generally positive pattern of results. Performance was substantially lower for detecting contextual representation for students who identify race as other (AUC_{CR} = 0.65) and for detecting outcome orientation for students who identify as belonging to two or more races (AUC_{OO} = 0.46). This relatively poor performance may be due to the small sample size of these constructs within these subgroups (i.e., 5 students identified as two or more races), or the fact that the other group likely represents a variety of different racial groups whose members aren't necessarily similar to each other. In future work, we hope to collect larger samples in order to ensure acceptable performance for these cases and examine the open responses for those who choose other.

Within the data, there were many students who declined to report gender (N = 9) and race (N = 19). Both groups who declined to report performed close to the average model performance, across groups and contexts.

To understand if there are any differences in student behaviors in using the platform between learners in different demographic groups, we examined whether the top features varied by gender and racial/ethnic groups, in terms of either mean or distribution. As shown in the Appendix, male and female students demonstrate similar behaviors, with one exception: female students on average tend to provide lengthier responses in the *Plan* phase (e.g., higher maximum value of the number of characters used) than male students. Students who did not specify their gender tend to have shorter answers and make fewer responses in general.

When comparing student behaviors across racial/ethnic groups, we found that African American students are likely to use more words and characters in their responses than Hispanic/Latinx or White students. For example, on average, African American students had a higher maximum number of characters used in the *Plan* phase and a higher average number of words used in the *Review* phase. However, high variation is also observed in this group (e.g., higher standard deviation of the number of characters used).

7. DISCUSSION

7.1. MAIN FINDINGS

Given the importance of self-regulation in learning, specifically in the problem-solving process, an increasing number of studies have looked into ways to promote self-regulated learning. This first requires the ability to accurately measure SRL so that interventions can be introduced to encourage and guide students to self-regulate effectively. However, the most common ways of measuring SRL in a fine-grained fashion – either through self-report and think-aloud protocols –are difficult to automate and scale, and they can also interrupt or interfere with the learning task. Log data collected from computer-based learning environments offer an unobtrusive and potentially scalable solution to help understand when and how students self-regulate within the problem-solving process, in order to inform decisions on intervention (e.g., Aleven et al., 2006). However, previous automated detection of SRL constructs using log data has mostly not been explicitly connected to SRL theory. In the current work, we explored the possibility of detecting SRL constructs at a fine-grained level, focusing on detecting cognitive operations (i.e., monitoring, assembling, rehearsing, and translating), outlined in the SMART model (Winne, 2017). Specifically, we detected the presence of six self-regulation indicators related to four categories of operations: 1) numerical representation, 2) contextual representation, 3) outcome orientation, 4) data transformation, 5) following plans, and 6) incorporating information. Detectors were built using a machine learning approach and were evaluated with a 10-fold student-level cross-validation. The detectors were found to be accurate and valid across demographic groups, with AUC ROC ranging from .76-.89.

To understand the detectors, feature importance was examined using SHAP values. The top five features with the highest absolute SHAP values were identified for each detector. With the features identified, we find that except for outcome orientation and following plans, the detectors primarily rely on features extracted from the *Understand* and *Plan* phases of the learning activity, the two phases where students assemble information and make plans. In particular, the numerical representation detector mainly relies on features that examine the numerical values used in the *Understand* phase as well as features that compare the similarity between student responses and the problem item. The numerical value feature makes sense, as the detector is operationalized to identify if numerical components are processed and represented when students assemble information.

However, the maximum similarity feature, a feature that takes both numerical values and text into account and examines how similar students' responses are to the problem item, also contributes to the NR indicator. This finding suggests that the NR detector not only examines if numbers are used in responses, but also how they are used in relation to the problem. As such, this finding validates the operationalization of this indicator, showing that the learner demonstrates a level of understanding of how numerical values are used in math problems, creating a representation of the problem space utilizing numbers.

The contextual representation detector looks at the keywords used in students' responses in the *Understand* phase and the length of the responses in the *Plan* phase, which indicates the relationship that the longer the responses are when a student is forming a plan, the more likely it is for the student to contextually represent the problem. When predicting the presence of outcome orientation, the model utilizes features extracted in the *Understand* and the *Review* phases, understanding students' use of keywords, nouns, and numerical values in these two phases. The data transformation detector checks the number of top strategies students select as well as the length and the variation in the length of the responses in the *Understand* and the *Plan* phase.

The features that predict students' behavior of following plans and incorporating information mainly come from the *Plan* phase, which coincide with the operationalization of the two indicators of which we expect to observe when students form plans. In specific, we find that the standard deviation of the number of verbs used in the *Plan* phase negatively predicts the following-plans behavior. Since the strategies selected (part of responses collected in the *Plan* phase) typically involve the use of verbs (e.g. draw a picture, or make a table), the more similar the response is in the planning journal (the task that asks students to write down the plan) in terms of the verb counts, the more likely it is for students to incorporate selected strategies into their plans, hence demonstrating the behavior of following-plans.

The incorporating information detector relies on features that examine the length of the responses (i.e., number of characters) and the use of pronouns in the *Plan* phase. A negative relationship is found between the length of the responses and the behavior of incorporating information. Though the relationship seems counterintuitive, as it is common to assume an extensive plan could be the result of incorporating information, after examining positive cases, we find that students who incorporate information tend to have straightforward and succinct plans, usually with a bullet list, outlining the steps that they will follow. In these plans, sentences often start with a first-person pronoun (e.g., "I will use" or "My plan is"), which possibly explain the positive relationship between the use of pronouns and the positive indication of the construct.

Additionally, we examined model performance on different demographic subgroups of students, both in terms of gender and racial/ethnic groups, to verify their fairness and lack of algorithmic biases. Relatively small differences were observed in each comparison, and no student group (either gender or racial/ethnic group) consistently had the best-performing detectors.

7.2. Applications

The detectors built in the current study provide two advances over the previous state-of-the-art in SRL detection. First, previous SRL detectors generally identified higher-level strategies and were not typically linked to theory; in contrast, our detectors were explicitly based on an SRL model in order to identify theoretically-grounded SRL constructs at a finer-grain size. Having developed these fine-grained models of behavior directly associated with the cognitive operations of SMART, we can conduct analyses to further our understanding of the role that cognitive operations play in the broader process of SRL. For example, we can investigate questions about how often students use these cognitive operations in each of the four tasks outlined in the Winne and Hadwin's four-stage model, and how the engagement and the frequency of the engagement in these cognitive operations contribute to the success of completing the tasks. Results from future analyses will help expand the current theoretical understanding on SRL, adding specificity to the still high-level processes represented in contemporary SRL theory.

Second, given that most previous detectors have not been connected explicitly to constructs in SRL theory, it has been difficult to use them with theory-driven interventions. The detectors proposed in the current study are developed based on a theoretical model of SRL (Winne, 2017; Winne and Hadwin, 1998) and are operationalized to capture key aspects of the cognitive operations in the model. These detectors can therefore be used to facilitate the development of adaptive learning environments that respond to student SRL, in a fashion connected to theory. For instance, a student demonstrating an outcome orientation could be encouraged to reflect further on their strategy, or a reminder could be provided when students fail to follow the plans they had previously formed.

Similarly, these detectors could provide teachers with information (e.g., through a dashboard) on how students are approaching problems. Such a dashboard could support teachers as they adapt their instruction and classroom strategies. For example, this data might allow teachers to provide more tailored instruction by grouping students based on their use of SRL strategies. Second, by examining aggregate data for a given problem (e.g., data from an entire class), teachers can gain insight on the properties of that problem that may inform instructional design. Problem items tend to differ in type and complexity, thus a different use of SRL reflected by a different pattern of engagement in the six SRL indicators may be expected. For example, students may be less likely to use outcome orientation in open-ended questions, or students may be more likely to use data transformation when a problem contains components and expressions that can be converted to equations. Knowing how students approach and respond to different problems, measured by SRL indicators, can aid teachers as they create, curate, or revise the items in future assignments. For example, teachers wishing to develop numerical representation skills for students, may select a problem that has a proven record of students using numerical representation in the past. As with any application of this nature, careful attention will be needed in design to ensure that data is presented in the most useful form for teachers and appropriately represents the uncertainty in the model (i.e., the degree of false positives or false negatives).

7.3. Limitations and future works

This work has five principal limitations that should be addressed in future work. First, when validating the fairness of the models, the sample size is small (less than five students) for several student groups. Reliable comparison of the model performance for these groups of students is therefore not possible. In future work, larger and more representative samples will need to be collected in order to validate model performance for a broader range of student groups. This is a general challenge in algorithmic bias work -- often the groups most at risk of algorithmic bias are insufficiently represented in the data to check for algorithmic bias (Baker and Hawn, 2022).

Second, although our detectors are based on a theoretical model of SRL, the operationalization of our constructs is contextualized in the current learning environment, so our detectors may be platform-specific. Future work should study the generalizability of the current detectors across platforms and explore how they can be adapted for use in other learning environments. To the extent that some of our detectors (such as the data transformation detector) apply across learning environments, we can investigate their performance within those contexts to evaluate their generalizability (see, for instance, Paquette and Baker, 2017).

Third, since the detectors are currently trained and modeled on complete *Thinklets*, they will have some limitations in the ways they can be used when being implemented in a learning platform. Specifically, the detectors will only be able to make predictions after a student has solved a problem, providing an indicator at that point on the student's use or lack of use of monitoring, assembling, rehearsing, and translating, in the problem-solving process. As such, these detectors will not provide immediate detection of these strategies when students are working through a problem. However, they can still be used to inform teachers and direct their feedback after a problem has concluded, in between problems or for the next problem. To enable other uses, it may be relevant to examine ways of also making early predictions based on incomplete *Thinklets* in order to provide detection during the problem-solving process, enabling real-time interventions.

Future work should also consider additional methods for ground truth labeling. In this work, we used a post-hoc tagging approach (through text replays), to identify indicators of SRL-related strategies. This approach has the potential to miss crucial "in-the-moment" events that are not evident from the log data alone. Future studies could examine how post-hoc tagging used in the current study aligns with in-the-moment tagging, reported either by student themselves or external observers/interviewers (e.g., Baker et al., 2004) to examine additional aspects of SRL.

Finally, future work should consider expanding the scope of this work. In the current study, seven constructs were identified and six modeled. SRL, as a process, covers a much broader range of behaviors and strategies that elicit the use of various cognitive operations. Future studies should model and detect a broader range of cognitive operations throughout the four stages of self-regulated learning in the context of problem-solving.

7.4. CONCLUSIONS

To better understand and facilitate the use of self-regulation in problem-solving, the current study tested the possibility of scaling up SRL measurement by leveraging machine learning to automatically detect individual SRL indicators through the lens of the SMART model. We built automated detectors that identify six commonly used strategies in math problem-solving, indicating four of the five operations outlined in the SMART model (namely, monitoring, assembling, rehearsing, and translating). Our detectors were found to be reliable and generalizable. These detectors were also tested on different student populations to verify their fairness and lack of algorithmic bias. Given these properties, we anticipate implementing the detectors in the learning environment to collect more fine-grained data and to leverage the detection to inform interventions, creating more positive experiences in mathematical problem-solving.

8. ACKNOWLEDGEMENTS

The research reported here was supported by the EF+Math Program of the Advanced Education Research and Development Program (AERDF) through funds provided to the University of Pennsylvania, University of Minnesota, and CueThink. The opinions expressed are those of the authors and do not represent views of the EF+Math Program or AERDF.

REFERENCES

AGUILAR, S.J., KARABENICK, S.A., TEASLEY, S.D., AND BAEK, C. 2021. Associations between learning analytics dashboard exposure and motivation and self-regulated learning. *Computers & Education 162*.104085

ALEVEN, V., MCLAREN, B., ROLL, I., AND KOEDINGER, K. 2006. Toward meta-cognitive tutoring: a model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education 16*, 2, 101–128.

ALEVEN, V., ROLL, I., MCLAREN, B.M., AND KOEDINGER, K.R. 2016. Help helps, but only so much: research on help seeking with intelligent tutoring systems. *International Journal of Artificial Intelligence in Education 26*, 1, 205–223.

AZEVEDO, R., JOHNSON, A., CHAUNCEY, A., GRAESSER, A., ZIMMERMAN, B., AND SCHUNK, D. 2011. Use of hypermedia to assess and convey self-regulated learning. *Handbook of Self-Regulation of Learning and Performance,* B. Zimmerman and D.H. Schunk, Eds. New York, NY: Routledge. 102–121.

AZEVEDO, R., TAUB, M., AND MUDRICK, N.V. 2017. Understanding and reasoning about real-time cognitive, affective, and metacognitive processes to foster self-regulation with advanced learning technologies. *Handbook of Self-Regulation of Learning and Performance*, D.H. Schunk and J.A. Greene, Eds. New York, NY: Routledge. 254–270.

BAKER, R.S. AND DE CARVALHO, A.M.J.A. 2008. Labeling student behavior faster and more precisely with text replays. *Proceedings of the 1st International Conference on Educational Data Mining*, R.S. Baker, T. Barnes, and J.E. Beck, Eds. International Educational Data Mining Society. 38–47.

BAKER, R.S., CORBETT, A.T., AND KOEDINGER, K.R. 2004. Detecting student misuse of intelligent tutoring systems. *International Conference on Intelligent Tutoring Systems*, S. Lester, R.M. Vicari, and F. Paraguacu, Eds. Springer, Berlin, Heidelberg. 531–540.

BAKER, R.S., CORBETT, A.T., AND WAGNER, A.Z. 2006. Human classification of low-fidelity replays of student actions. *Proceedings of the Educational Data Mining Workshop at the 8th International Conference on Intelligent Tutoring Systems*, M. Ikeda, K. Ashley, and TW. Chan Eds. Springer, Berlin, Heidelberg. 29–36.

BAKER, R.S. AND HAWN, A. 2022. Algorithmic bias in education. *International Journal of Artificial Intelligence in Education*, 32(4), 1052-1092.

BAKER, R.S., MITROVIĆ, A., AND MATHEWS, M. 2010. Detecting gaming the system in constraint-based tutors. *Proceedings of the 18th Annual Conference on User Modeling, Adaptation, and Personalization*, P. Bra, A. Kobsa, and D. Chin, Eds, Springer, Berlin, Heidelberg. 267–278.

BAKER, R.S. AND OCUMPAUGH, J. 2016. Interaction-based affect detection in educational software. *The Oxford Handbook of Affective Computing*, R. Calvo, S. D'Mello, J. Gratch, and A. Kappas, Eds. New York, NY: Oxford University Press. 233–245.

BAKER, R.S. AND YACEF, K. 2009. The state of educational data mining in 2009: a review and future visions. *Journal of Educational Data Mining 1*, 1, 3–17.

BANDURA, A. 1986. *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs, NJ: Prentice Hall.

BISWAS, G., BAKER, R.S., AND PAQUETTE, L. 2017. Data mining methods for assessing self-regulated learning. *Handbook of Self-Regulation of Learning and Performance*, D.H. Schunk and J.A. Greene, Eds. New York, NY: Routledge. 388–403.

BISWAS, G., JEONG, H., KINNEBREW, J.S., SULCER, B., AND ROSCOE, R. 2010. Measuring self-regulated learning skills through social interactions in a teachable agent. *Research and Practice in Technology Enhanced Learning 05*, 02, 123–152.

BOEKAERTS, M. 1999. Self-regulated learning: where we are today. *International Journal of Educational Research 31*, 6, 445–457.

BOSCH, N., ZHANG, Y., PAQUETTE, L., BAKER, R., OCUMPAUGH, J., AND BISWAS, G. 2021. Students' verbalized metacognition during computerized learning. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, ACM. 1–12.

BOTELHO, A.F., BAKER, R.S., OCUMPAUGH, J., AND HEFFERNAN, N.T. 2018. Studying affect dynamics and chronometry using sensor-free detectors. *Proceedings of the 11th International Conference on Educational Data Mining*, K.E. Boyer and M. Yudelson, Eds. International Educational Data Mining Society. 157–166.

CHARMAZ, K. 1983. The grounded theory method: An explication and interpretation. *Contemporary Field Research*, R. Emerson, Eds. Boston, MA: Little, Brown and Company. 109–126.

CHEN, T. AND GUESTRIN, C. 2016. XGBoost: a scalable tree boosting system. *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining*, ACM. 785–794.

CHO, M.-H. AND YOO, J.S. 2017. Exploring online students' self-regulated learning with self-reported surveys and log files: a data mining approach. *Interactive Learning Environments 25*, 8, 970–982.

CLEARY, T.J. AND CHEN, P.P. 2009. Self-regulation, motivation, and math achievement in middle school: Variations across grade level and math context. *Journal of School Psychology 47*, 5, 291–314.

CONATI, C., PORAYSKA-POMSTA, K., AND MAVRIKIS, M. 2018. AI in education needs interpretable machine learning: lessons from open learner modelling. http://arxiv.org/abs/1807.00154.

DEVER, D.A., AMON, M.J., VRZAKOVA, H., WIEDBUSCH, M.D., CLOUDE, E.B., AND AZEVEDO, R. 2022. Capturing sequences of learners' self-regulatory interactions with instructional material during game-based learning using auto-recurrence quantification analysis. *Frontiers in Psychology, 13*, 813677.

DEVOLDER, A., VAN BRAAK, J., AND TONDEUR, J. 2012. Supporting self-regulated learning in computer-based learning environments: systematic review of effects of scaffolding in the domain of science education. *Journal of Computer Assisted Learning 28*, 6, 557–573.

DICERBO, K.E. AND KIDWAI, K. 2013. Detecting player goals from game log files. *Proceedings of the 6th International Conference on Educational Data Mining,* S.K. D'Mello, R.A. Calvo, and A. Olney, Eds. International Educational Data Mining Society. 314-315.

EFKLIDES, A. 2011. Interactions of metacognition with motivation and affect in self-regulated learning: the MASRL model. *Educational Psychologist 46*, 1, 6–25.

FARHANA, E., POTTER, A., RUTHERFORD, T., AND LYNCH, C. F. 2021. Feedback and self-regulated learning in science reading. *Proceedings of the 14th International Conference on Educational Data Mining,* I.-H. Hsiao, S. Sahebi, F. Bouchet, and J.-J.Vie, Eds. International Educational Data Mining Society. 820-826

GARDNER, J., BROOKS, C., AND BAKER, R. 2019. Evaluating the fairness of predictive student models through slicing analysis. *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, D. Azcona and R. Chung, Eds. ACM. 225–234.

GRAESSER, A. AND MCNAMARA, D. 2010. Self-regulated learning in learning environments with pedagogical agents that interact in natural language. *Educational Psychologist 45*, 4, 234–244.

GRAESSER, A.C., PENUMATSA, P., VENTURA, M., CAI, Z., AND HU, X. 2007. Using LSA in autotutor: learning through mixed initiative dialogue in natural language. *Handbook of Latent Semantic Analysis*, T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, Eds. Mahwah: Erlbaum. 243–262.

GREENE, J.A., DEEKENS, V.M., COPELAND, D.Z., AND YU, S. 2017. Capturing and modeling self-regulated learning using think-aloud protocols. *Handbook of Self-Regulation of Learning and Performance*, D.H. Schunk and J.A. Greene, Eds. New York, NY: Routledge. 323–337.

GREENE, J.A., ROBERTSON, J., AND COSTA, L.-J.C. 2013. Assessing self-regulated learning using think-aloud methods. *Handbook of Self-Regulation of Learning and Performance,* B.J. Zimmerman and D.H. Schunk, Eds. New York, NY: Routledge. 313–328.

KARUMBAIAH, S. AND BROOKS, J. 2021. How colonial continuities underlie algorithmic injustices in education. *Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology*, IEEE, 1–6.

KINNEBREW, J.S., LORETZ, K.M., AND BISWAS, G. 2013. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining 5*, 1, 190–219.

KIZILCEC, R.F. AND LEE, H. 2020. Algorithmic fairness in education. *The Ethics of Artificial Intelligence in Education*, W. Holmes and K. Porayska-Pomsta, Eds. Taylor & Francis. 174–202.

KÖCK, M. AND PARAMYTHIS, A. 2011. Activity sequence modelling and dynamic clustering for personalized e-learning. *User Modeling and User-Adapted Interaction 21*, 1–2, 51–97.

KOVANOVIC, V., GAŠEVIĆ, D., DAWSON, S., JOKSIMOVIC, S., AND BAKER, R. 2016. Does time-on-task estimation matter? Implications on validity of learning analytics findings. *Journal of Learning Analytics 2*, 3, 81–110.

LABUHN, A.S., ZIMMERMAN, B.J., AND HASSELHORN, M. 2010. Enhancing students' self-regulation and mathematics performance: the influence of feedback and self-evaluative standards. *Metacognition and Learning 5*, 2, 173–194.

LEE, D.M.C., RODRIGO, MA.M.T., BAKER, R.S.J. D., SUGAY, J.O., AND CORONEL, A. 2011. Exploring the relationship between novice programmer confusion and achievement. *International Conference on Affective Computing and Intelligent Interaction*, S. D'Mello, A. Graesser, B. Schuller, J.C. Martin, Eds. Springer, Berlin, Heidelberg.175–184.

LUNDBERG, S.M., ERION, G.G., AND LEE, S.I. 2019. Consistent individualized feature attribution for tree ensembles. *arXiv:1802.03888*.

MU, T., JETTEN, A., AND BRUNSKILL, E. 2020. Towards suggesting actionable interventions for wheel-spinning students. *Proceedings of the 13th International Conference on Educational Data Mining*, A.N. Rafferty, J. Whitehill, V. Cavalli-Sforza, and C. Romero, Eds. International Educational Data Mining Society. 183–193.

MULDNER, K., BURLESON, W., VAN DE SANDE, B., AND VANLEHN, K. 2011. An analysis of students' gaming behaviors in an intelligent tutoring system: predictors and impacts. *User Modeling and User-Adapted Interaction 21*, 1–2, 99–135.

NOTA, L., SORESI, S., AND ZIMMERMAN, B.J. 2004. Self-regulation and academic achievement and resilience: A longitudinal study. *International Journal of Educational Research 41*, 3, 198–215.

OCUMPAUGH, J., BAKER, R., GOWDA, S., HEFFERNAN, N., AND HEFFERNAN, C. 2014. Population validity for educational data mining models: A case study in affect detection. *British Journal of Educational Technology 45*, 3, 487–501.

OCUMPAUGH, J., HUTT, S., ANDRES, J.M.A.L., BAKER, R.S., BISWAS, G., BOSCH, N., PAQUETTE, L., AND MUNSHI, A. 2021. Using qualitative data from targeted interviews to inform rapid AIED development. *Proceedings of the 29th International Conference on Computers in Education*, M.M.T. Rodrigo, S. Iyer, A. Mitrovic, H.N.H. Cheng, D. Kohen-Vacs, C.M.A. Palalas, R. Rajenran, K. Seta, and J. Wang, Eds. Asia-Pacific Society for Computers in Education. 69–74.

OKUR, E., ASLAN, S., ALYUZ, N., ARSLAN ESME, A., AND BAKER, R.S. 2018. Role of socio-cultural differences in labeling students' affective states. *Proceedings of the 19th International Conference on Artificial Intelligence in Education*, C.P. Rose, R, Martinez-Maldonado, H.U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, B. McLaren, B. du Boulay, Eds. Springer, Cham. 367–380.

PANADERO, E. 2017. A review of self-regulated learning: six models and four directions for research. *Frontiers in Psychology 8*, 422.

PAQUETTE, L. AND BAKER, R.S. 2017. Variations of gaming behaviors across populations of students and across learning environments. *Proceedings of the 18th International Conference on Artificial Intelligence in Education*, E. André, R. Baker, X. Hu, M.M.T. Rodrigo, and B. du Boulay, Eds. Springer Cham. 274–286.

PAQUETTE, L. AND BAKER, R.S. 2019. Comparing machine learning to knowledge engineering for student behavior modeling: a case study in gaming the system. *Interactive Learning Environments 27*, 5–6, 585–597.

PAQUETTE, L., DE CARVALHO, A.M.J.A., AND BAKER, R.S. 2014. Towards understanding expert coding of student disengagement in online learning. *Proceedings of the 36th Annual Cognitive Science Conference*, P. Bello, M. Guarini, M. McShane, and B. Scassellati, Eds. Cognitive Science Society. 1126–1131.

PAQUETTE, L., OCUMPAUGH, J., LI, Z., ANDRES, A., AND BAKER, R. 2020. Who's learning? Using demographics in EDM research. *Journal of Educational Data Mining 12*, 3, 1–30.

PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., ET AL. 2011. Scikit-learn: Machine Learning in Python. *Journal of machine Learning research 12*, 2825–2830.

PINTRICH, P.R. 2000. The role of goal orientation in self-regulated learning. *Handbook of Self-Regulation*, M. Boekaerts, P. Pintrich, and M. Zeidner, Eds. San Diego, CA: Academic Press. 451–502.

PINTRICH, P.R., SMITH, D., GARCIA, T., AND MCKEACHIE, W. 1991. A manual for the use of the motivated strategies for learning questionnaire (MSLQ).

RICHEY, J. E., ZHANG, J., DAS, R., ANDRES-BRAY, J. M., SCRUGGS, R., MOGESSIE, M., BAKER, R.S, AND MCLAREN, B. M. 2021. Gaming and confrustion explain learning advantages for a math digital learning game. *Proceedings of the 22nd International Conference on Artificial Intelligence in Education,* G. Biswas, S. Bull, J. Kay, and A. Mitrovic, Eds. Springer, Berlin, Heidelberg. 342–355.

ROLL, I., ALEVEN, V., MCLAREN, B.M., AND KOEDINGER, K.R. 2007. Can help seeking be tutored? Searching for the secret sauce of metacognitive tutoring. *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, R. Luckin, K.R. Koedinger, and J. Greer, Eds. IOS Press. 203-210.

ROTH, A., OGRIN, S., AND SCHMITZ, B. 2016. Assessing self-regulated learning in higher education: a systematic literature review of self-report instruments. *Educational Assessment, Evaluation and Accountability 28*, 3, 225–250.

SABOURIN, J., SHORES, L.R., MOTT, B.W., AND LESTER, J.C. 2012. Predicting student self-regulation strategies in game-based learning environments. *Proceedings of the 11th International Conference on Intelligent Tutoring Systems*, S.A. Cerri, W.J. Clancey, G. Papadourakis, and K. Panourgia, Eds. Springer Berlin Heidelberg. 141–150.

SAO PEDRO, M.A., DE BAKER, R.S.J., GOBERT, J.D., MONTALVO, O., AND NAKAMA, A. 2013. Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction 23*, 1, 1–39.

SCHOOLER, J.W., OHLSSON, S., AND BROOKS, K. 1993. Thoughts beyond words: when language overshadows insight. *Journal of Experimental Psychology: General 122*, 2, 166–183.

SCULLEY, D., SNOEK, J., WILTSCHKO, A., AND RAHIMI, A. 2018. Winner's curse. *On pace, progress, and empirical rigor*. *Proceedings of the 6th International Conference on Leaning Representations*, Y. Bengio and Y. LeCun, Eds. OpenReview.

SEGEDY, J.R., KINNEBREW, J.S., AND BISWAS, G. 2015. Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *Journal of Learning Analytics 2*, 1, 13–48.

SMITH, T.F. AND WATERMAN, M.S. 1981. Identification of common molecular subsequences. *Journal of Molecular Biology 147*, 1, 195–197.

SWAMY, V., RADMEHR, B., KRCO, N., MARRAS, M., AND KÄSER, T. 2022. Evaluating the explainers: black-box explainable machine learning for student success prediction in MOOCs. *Proceedings of the 15th International Conference on Educational Data Mining*, A. Mitrovic and N. Bosch, Eds. International Educational Data Mining Society. 98-109.

TAUB, M., AZEVEDO, R., BOUCHET, F., AND KHOSRAVIFAR, B. 2014. Can the use of cognitive and metacognitive self-regulated learning strategies be predicted by learners' levels of prior knowledge in hypermedia-learning environments? *Computers in Human Behavior 39*, 356–367.

WALONOSKI, J.A. AND HEFFERNAN, N.T. 2006. Prevention of off-task gaming behavior in intelligent tutoring systems. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, M. Ikeda, K.D. Ashley, and TW. Chan, Eds. Springer Berlin Heidelberg, 722–724.

WEBB, M.E., FLUCK, A., MAGENHEIM, J., MALYN-SMITH, J., WATERS, J., DESCHENES, M., AND ZAGAMI, J. 2021. Machine learning for human learners: opportunities, issues, tensions and threats. *Educational Technology Research and Development 69*, 4, 2109–2130.

WEINSTEIN, C., PALMER, D., AND SCHULTE, A.C. 1987. Learning and study strategies inventory (LASSI). *FL: H & H Publishing*.

WESTON, C., GANDELL, T., BEAUCHAMP, J., MCALPINE, L., WISEMAN, C., AND BEAUCHAMP, C. 2001. Analyzing interview data: the development and evolution of a coding system. *Qualitative Sociology 24*, 3, 381–400.

WIJFFELS, J., STRAKA, M., AND STRAKOVA, J. 2017. Package "udpipe".

WINNE, P.H. 1997. Experimenting to bootstrap self-regulated learning. *Journal of Educational Psychology 89*, 3, 397–410.

WINNE, P.H. 2004. Students' calibration of knowledge and learning processes: Implications for designing powerful software learning environments. *International Journal of Educational Research 41*, 6, 466–488.

WINNE, P.H. 2005. Key issues in modeling and applying research on self‐regulated learning. *Applied Psychology 54*, 2, 232–238.

WINNE, P.H. 2010a. Bootstrapping learner's self-regulated learning. *Psychological Test and Assessment Modeling 52*, 4, 472.

WINNE, P.H. 2010b. Improving measurements of self-regulated learning. *Educational Psychologist 45*, 4, 267–276.

WINNE, P.H. 2017. Learning analytics for self-regulated learning. *Handbook of Learning Analytics*, C. Lang, G. Siemens, A. Wise, D. Gašević, Eds. Society for Learning Analytics and Research. 531–566.

WINNE, P.H. AND HADWIN, A.F. 1998. Studying as self-regulated learning. *Metacognition in Educational Theory and Practice*, D.J. Hacker, J. Dunlosky, and A.C. Graesser, Eds. Hillsdale, NJ: Erlbaum. 277–304.

WINNE, P.H. AND PERRY, N.E. 2000. Measuring self-regulated learning. *Handbook of Self-Regulation,* M. Boekaerts, P. Pintrich, and M. Zeidner, Eds. Orlando, FL: Academic Press. 531–566.

WINNE, P.H., TENG, K., CHANG, D., LIN M.P., MARZOUK, Z., NESBIT, J.C., PATZAK, A., RAKOVIC, M., SAMADI, D., VYTASEK, J. 2019. nStudy: software for learning analytics about processes for self-regulated learning. *Journal of Learning Analytics 6*, 2, 95–106.

ZIMMERMAN, B.J. 1990. Self-regulated learning and academic achievement: an overview. *Educational Psychologist 25*, 1, 3–17.

ZIMMERMAN, B.J. 2000. Attaining self-regulation: a social cognitive perspective. *Handbook of Self-Regulation*, M. Boekaerts, P. Pintrich, and M. Zeidner, Eds. Orlando, FL: Academic Press. 13–39.

ZIMMERMAN, B.J. AND SCHUNK, D.H. 2011. Self-regulated learning and performance. *Handbook of Self-Regulation of Learning and Performance*, B. J. Zimmerman and D. H. Schunk, Eds. New York, NY: Routledge. 1–12.

Appendix

The Top Five Features from each Detector and their Average Values.

			Gender			Race/Ethnicity					
Feature Phase	Feature	All students (N = 72)	Male (33)	Female (28)	Left Blank (9)	African American (6)	Hispanic/ Latinx (18)	White (8)	Prefer Not to Say (19)	Other (14)	Two or more races (5)
Numerical Representation											
Understand	Mean N of responses that give numerical values	0.5	0.5	0.5	0.5	0.4	0.5	0.6	0.6	0.5	0.3
Understand	Max value of the similarity feature	1.0	1.1	1.0	1.0	0.9	1.0	1.2	1.1	1.2	1.0
Understand	SD of the similarity feature	0.4	0.4	0.4	0.4	0.4	0.3	0.4	0.3	0.4	0.4
Understand	Total N of responses	5.9	6.1	6.0	4.9	5.3	5.7	5.9	5.1	7.1	6.3
Plan	Avg value of the similarity feature	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Contextual Representation											
Understand	N of responses to the "what do you notice" question?	3.6	3.8	3.8	2.7	3.1	3.3	3.8	3.1	4.9	3.8
Understand	Avg N of keywords used	0.7	0.7	0.7	0.7	0.8	0.7	0.6	0.8	0.7	0.4
Understand	SD of the N of characters used	20.0	17.8	22.0	21.4	22.6	19.5	19.6	18.8	22.0	18.7
Thinklet	Total N of responses	13.6	13.6	14.6	9.7	12.6	14.6	13.1	11.8	15.4	13.0
Plan	Max value of the N of characters used	152.4	149.0	181.7	78.2	217.8	153.5	172.4	95.9	176.3	121.9
Outcome Orientation											
Review	Avg N of keywords used	0.1	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.2
Review	Avg N of words used	4.4	3.9	5.5	3.4	6.7	4.5	2.7	3.8	5.0	4.7
Understand	SD of the N of numerical values used	0.9	0.9	0.9	0.7	1.1	1.0	0.7	0.7	1.0	0.6
Understand	Is there a response to the "what is your estimated answer" question?	0.9	0.9	0.9	0.9	0.9	0.9	0.8	0.9	0.9	1.0
Review	Avg N of nouns used	5.4	5.4	5.6	4.3	5.6	5.7	5.1	5.0	5.6	5.3
Data Transformation											
Plan	N of common strategies selected	1.1	0.9	1.2	1.3	1.0	1.2	1.1	1.2	1.1	1.0
Understand	SD of the similarity feature	0.4	0.4	0.4	0.4	0.4	0.3	0.4	0.3	0.4	0.4
Plan	SD of the N of characters used	71.8	78.0	79.3	35.8	107.7	75.7	83.2	43.7	75.7	49.1
Understand	SD of the N of nouns used	1.1	1.0	1.2	1.1	1.2	1.0	1.0	1.0	1.1	1.2
Plan	Max value of N of words used	30.3	29.7	36.1	16.0	43.0	29.9	34.8	19.4	35.1	24.6
Following Plans											
Plan	SD of the N of verbs used	3.2	3.5	3.2	2.7	4.3	3.2	2.8	2.6	3.2	2.8
Plan	SD of the N of pronouns used	2.2	2.5	2.5	1.0	3.2	2.5	2.4	1.3	2.3	1.8

Review	Avg N of pronouns used	0.7	0.7	0.9	0.5	1.1	0.8	0.6	0.5	0.8	0.7
Plan	Avg N of verbs used	4.4	4.4	4.8	3.4	5.7	4.2	4.3	3.3	5.1	3.7
Review	Avg N of verbs used	1.8	1.8	2.0	1.4	2.2	1.9	1.7	1.6	1.9	1.6
Incorporating Information											
Plan	Avg N of characters used	84.7	87.0	97.5	45.9	116.0	81.1	98.6	53.3	102.9	54.3
Plan	Avg N of pronouns used	2.0	2.1	2.2	0.9	2.7	1.9	2.4	1.1	2.3	1.1
Plan	Max value of N pronouns used	4.1	4.1	4.8	1.8	5.8	4.2	4.6	2.4	4.6	3.4
Plan	Max value of N words used	30.3	29.7	36.1	16.0	43.0	29.9	34.8	19.4	35.1	24.6
Plan	SD of N of nouns used	2.4	2.8	2.3	1.7	3.5	2.4	2.0	1.9	2.4	1.6