Towards Suggesting Actionable Interventions for Wheel-Spinning Students

Tong Mu
Stanford University
tongm@cs.stanford.edu

Andrea Jetten
War Child Holland
Andrea.Jetten@warchild.nl

Emma Brunskill
Stanford University
ebrun@cs.stanford.edu

ABSTRACT

In some computerized educational systems, there is evidence of students *wheel-spinning*, where a student tries and repeatedly fails at an educational task for learning a skill. This may be particularly concerning in low resource settings. Prior research has focused on predicting and modeling wheel-spinning, but there has been little work on how to best help students stuck in wheel-spinning. We use past student system interaction data and a minimal amount of expert input to automatically inform individualized interventions, without needing experts to label a large dataset of interventions. Our method trains a model to predict wheel-spinning and utilizes a popular tool in interpretable machine learning, Shapley values, to provide individualized credit attribution over the features of the model, including actionable features like possible gaps in prerequisites. In simulation on two different statistical student models, our approach can identify a correct intervention with over 80% accuracy before the simulated student begins the activity they will wheel spin on. In our real dataset we show initial qualitative results that our proposed interventions match what an expert would prescribe.

Keywords

Explainable Machine Learning, Inferring Interventions, Wheel-Spinning, Feature Attribution, Shapley Values

1. INTRODUCTION

Educational technology is increasingly used in a wide array of K-12 settings and some students struggle. Beck et al. [6] coined the term "wheel-spinning" to denote students that were repeatedly trying, and failing, to successfully complete a specific skill after many attempts in an intelligent tutoring system. They additionally found it was a significant issue in two popular computerized educational systems. Such long repeated failures are likely to be an inefficient use of time for students, and may additionally contribute to lack of motivation for future learning.

Tong Mu, Andrea Jetten and Emma Brunskill "Towards Suggesting Actionable Interventions for Wheel Spinning Students" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 183 - 193

Although expert human instructors are often very good at diagnosing and assisting students who are stuck, it is time consuming for both the instructors and the students waiting for the instructor's intervention. Additionally many educational settings lack a sufficient number of expert teachers. Our research is particularly motivated by a collaboration with the non-profit War Child Holland whose program Can't Wait to Learn (CWTL) provides self-paced educational software on tablets primarily to children in or coming from conflict-affected regions. In such settings, a limited number of teachers must often address the learning needs of a large number of students with a wide variety of educational backgrounds. To give a specific example, in the classes in Uganda the program is implemented in, the average class size is 114 students per teacher. Additionally for some population of students where education is especially hard to access, the program is run by facilitators who do not have the same expertise as instructors to provide learning support for individuals. Methods that can automatically identify individualized interventions, such as having the student practice an activity to review a prerequisite skill, to help wheel-spinning students could be greatly beneficial for students and teachers. However, since the term was coined, there has been much work for modelling and predicting wheel-spinning [11, 12, 16], but little work in developing interventions.

There are many possible reasons a student may wheel-spin, including lack of required prior knowledge, a long gap in learning of the material, or an ineffective educational activity. One approach could be to have experts label a large dataset with expert prescribed interventions and train a model to predict those interventions. However in many cases the time necessary to label such a dataset can be infeasibly large. For example, in our real world dataset a domain expert needed 30 minutes to label the 6 wheel-spinning cases we use for a qualitative evaluation. This would translate to 120 expert hours to label our whole dataset of more than 2000 wheel-spinning cases.

In this work we present a method to automatically predict when an intervention could be helpful and which intervention to give. Our method uses prior student system log data and only requires a few hours of expert input. Our method takes as input a set of features, a subset of which are actionable and correspond to a concrete intervention (for example, the feature "prerequisite performance" could correspond to an intervention of reviewing that prerequisite). We then use featurized past student data to train a model to predict wheel-spinning. With the prediction model, we use methods from explainable machine for providing feature credit attribution for the prediction of individual datapoints, specifically Shapley values [19], to determine which actionable feature contributed most to the prediction and suggest an intervention.

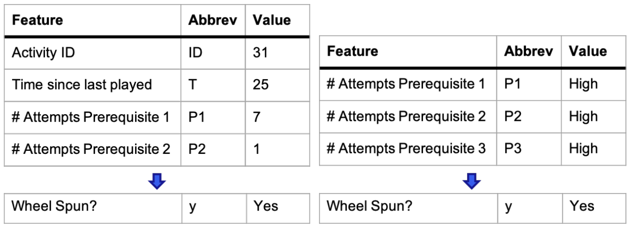
We evaluate the ability of our method to suggest correct interventions through simulation and through a qualitative study with our real data. Evaluating if our method is impactful will eventually require experimental studies. The costs of an experiment are high in our situation where this educational technology is being used by children in conflict-affected areas and who may be in remote villages without internet. Before embarking on such an effort, in this work we first assess the potential benefits and performance of our method. In simulation studies, we simulated students using two different student models, both based on the popular Bayesian Knowledge Tracing (BKT) [10] student model. In both of our simulations, our method can prescribe a correct action (a helpful intervention or correctly identifying no intervention is needed) with high accuracy before attempting an activity. This accuracy can be improved if the prediction is made at a later attempt. In an initial qualitative assessment in our real world CWTL setting we show our method's explanations are consistent with what an expert would prescribe in a majority of the cases. In the other cases the method did not have access to key data used by the expert, suggesting our method is able to identify correct interventions over correctly defined inputs.

Our method is, to our knowledge, one of the first works for both addressing automatically identifying interventions for wheel-spinning students and using interpretable machine learning in educational technologies. These results suggest that our method can help inform interventions, whether for carefully designed human-in-the-loop systems (such as only informing the teacher if confident the teacher is the best source) or for automated systems (jumping back to practice an earlier skill), and may help further adaptive automated systems for effective, efficient and engaging education.

2. RELATED WORKS

2.1 Wheel-Spinning

The term *wheel-spinning* was first coined by Beck et. al [6] where they examine its prevalence in two educational systems. Gong et al.[11] further explored models to predict wheel-spinning. Beck et al [5] found it applied to students in non-western societies as well. They also examined the influence of affective factors, and found it correlated with gaming the system. Matsuda et al. [16] examined using neural networks together with the BKT model [10] to predict wheel-spinning using only past student performance information. Kai et al. [12] investigate using decision trees to distinguish between productive persistence and wheel-spinning. Zhang et al. [24] make a comparison over many methods for detecting wheel-spinning. Wan et. al [23] take a step in modeling with actionable results by examining the effects of using prerequisite performance as features. They modeled wheel-spinning using both the average prerequisite performance and the weakest prerequisite and found that prerequisite knowledge was a reliable predictor of wheel-spinning and slightly improved model performance. In our work we propose a method to not only predict wheel-spinning, but also give suggestion of a possible intervention. We achieve this by designing our features to be actionable, such as incorporating performance on all prerequisites as separate features, with methods from explainable machine learning.



Feature	Abbrev	Value
Activity ID	ID	31
Time since last played	T	25
# Attempts Prerequisite 1	P1	7
# Attempts Prerequisite 2	P2	1
Wheel Spun?	y	Yes

(a) Example 1: Fake simplified datapoint inspired by CWTL

Feature	Abbrev	Value
# Attempts Prerequisite 1	P1	High
# Attempts Prerequisite 2	P2	High
# Attempts Prerequisite 3	P3	High
Wheel Spun?	y	Yes

(b) Example 2: Simple Binary Example

Figure 1: Simulated Student Setting

2.2 Explainable Machine Learning

Explainable Machine Learning is a rapidly growing popular field in the machine learning community. One subfield is the study of feature attribution which are methods that return how much each feature contributed to the total prediction of a datapoint in a machine learning model. In our work we use Shapely values [19], and the python implementation SHAP [13, 14] package to inform interventions. Shapley values are a method originating in game theory for fairly allocating a payout between participants. It has recently found popularity in explainable machine learning to calculate feature attributions. Shapley values have been used widely both within and outside of machine learning, including in medical applications [15], social network node analysis [17], and studying carbon emission quotas in China [25]. To our knowledge this is one of the first works on using Shapley values and explainable machine learning methods for educational technologies.

3. METHODS

In this section we present an algorithm to help students likely to wheel spin by suggesting actionable interventions. Our goal is to provide a method for using past student log data to predict when and which intervention a student will need to prevent wheel-spinning. We would also like to minimize interruptions to student-activity pairs who do not wheel spin. Similar to prior work [6] we define wheel-spinning as when a student consecutively fails an educational activity more than a threshold number of times. We will refer to the student-activity pair of the i^{th} student working on the j^{th} activity as $pair_{ij}$. To achieve our goal, for every $pair_{ij}$, our algorithm uses a 2 level decision process shown in Algorithm 1.

We first train a machine learning model using an existing dataset of student log data to predict wheel-spinning. Our overall algorithm (Algorithm 1) is compatible with any machine learning model that outputs probabilities of wheel-spinning given an input set of student features. In our work we use the popular gradient boosting method XGBoost [9]. When a student is using the educational program, given their current state the algorithm uses the trained model to predict if a student-activity pair will result in wheel-spinning. We define the number of failed attempts a student makes before we decide to possibly suggest an intervention as n. If the student has reached the n^{th} attempt on the current item our method uses the wheel-spinning model to predict if wheel-spinning will occur. If the output probability of wheel-spinning of the model is greater than a threshold, p, the algorithm will then propose a potential intervention. n and p are hyperparameters, and we provide further discussion on their effect in Sections 4.5 and 6.

Interventions are proposed using a method of feature attribution from explainable machine learning, Shapley Values[19] (described in more detail in Section 3.1). We use Shapley values to assign a contribution value to each feature used in the wheel-spinning prediction model. A subset of these features are designed to be actionable and correspond to an intervention. For example, Figure 1a shows example feature values of a fake datapoint for a student-activity pair inspired by CWTL. An example of an actionable feature in this fake datapoint is number of attempts required on a prerequisite skill. If assigned a high positive attribution value, it would suggest the student needs more practice on that prerequisite. Our method identifies the actionable feature with the highest Shapley value and suggests the corresponding intervention to give to the student. Non-actionable features that do not correspond to an intervention but increase prediction accuracy are also included.

There are a few places that require expert input, for example choosing hyperparameters n and p and designing the features and interventions. For experiments with our real world dataset, we worked together with a domain expert to create actionable features and corresponding interventions.

3.1 Background on Shapley Values

In this section we provide some background on the calculation and properties of Shapley Values [19] which is used in our method to provide feature attribution for the wheel-spinning prediction of individual datapoints. Shapley values originated in game theory and in the context of explainable machine learning, provide an attribution for how much each feature contributes to the total prediction of a datapoint. To give an example, consider a setting where we are predicting wheel-spinning using features in our dataset. We will refer to this setting as example setting 1. The datapoint in Figure 1a gives an example datapoint in this setting. Assume the mean prediction of the wheel-spinning model over all the datapoints in this example is 0.5, and for this datapoint the model predicts a probability of 0.8, which is +0.3 from the mean. The Shapley values for each feature give the contribution of each feature to this difference from the mean where the sum of contributions over all features must be +0.3. For example the features ID, T, $P2$ could all be attributed -0.1 and the feature $P1$ could be attributed +0.6. This attribution would suggest the value of the number of attempts on prerequisite 1 is likely to be responsible for the increased probability of wheel-spinning over the average wheel-spinning prediction.

Shapley values is the only method for attribution that satisfies the following desirable properties which together are the definition of a fair attribution[19]: symmetry (two features that contribute equally will have the same value), dummy (a feature that does not change the prediction has a value of 0), and additivity (if the prediction model is the sum of multiple models, the value of a feature in the prediction model is the sum of all values over the individual models). To give intuition of why the symmetry and dummy properties are desirable in this context, consider a second, simpler example setting, example setting 2, where we are also predicting wheel-spinning but all inputs and outbuts are binary. In this setting the wheel-spinning prediction is for one activity that is thought to have three prerequisites ($P1$, $P2$, $P3$). Figure 1b gives an example datapoint in this setting. Consider the case where two prerequisites, $P1$ and $P2$, are equally important and $P3$ was incorrectly labeled as a prerequisite and its value never influences the prediction of the model. Because $P1$ and $P2$ are equally important and for the datapoint in figure 1b both their values are high, we would like them to have equal attribution, or to satisfy the "symmetry" property. Additionally, because $P3$ was incorrectly labelled as a prerequisite we would want it to be given 0 attribution regardless of its value, or to satisfy the "dummy" property.

We now describe formally how to calculate Shapley values. Let $\mathcal{F}$ denote the set of features and X denote the dataset. One example of a datapoint in X from example setting 1 is the example datapoint Figure 1a, which we will refer to as x_i. In this example $\mathcal{F} = \{ID,\ T,\ P1,\ P2\}$. Also assume there is a function V where $V(x_i)$ is the predicted value on datapoint x_i. Let $SHAP(x_i, f_j)$ be x_i's Shapley value for feature f_j. In our example, $V(x_i)$ would output the probability of x_i wheel-spinning. For a subset of features s, ($s \subseteq \mathcal{F}$) we define a fake datapoint, $x_{i,s}$, as a datapoint that only includes the the values of x_i for the features in s. In our example, one potential s could be $\{T, P1\}$, and the corresponding $x_{i,s} = [T : 25, P1 : 7]$. For a feature f_j, we define a coalition of features, F, as a subset of $\mathcal{F}$ that does not include f_j. We define $\mathcal{C}$ as the set of all unique coalitions for f_j and let F_k denote the k^{th} coalition in this set. Let the contribution of f_j in coalition F_k to the prediction of x_i be the difference in prediction of the datapoint without f_j and the datapoint with f_j included, or $V(x_{i,F_k \cup f_j}) - V(x_{i,F_k})$. In our example, the $s = \{T, P1\}$ is a coalition of features for $f_j = ID$ as it does not include f_j. If the probability of wheel-spinning on $x_{i,s}$ ($V([T\text{: } 25,\ P1\text{: } 7])$) is -0.1 and $V(x_{i,F_k \cup f_j}) = V([\textbf{ID: 31}, T : 25, P1 : 7])$ = -0.2. Then the difference $V(x_{i,F_k \cup f_j}) - V(x_{i,F_k})$ = -0.1

The Shapley value is then the expected contribution of f_j averaged over all coalitions:

$$SHAP(x_i, f_j) = \mathbb{E}_C[V(x_{i,F \cup f_j}) - V(x_{i,F})] \tag{1}$$

$$= \sum_{F_k \in C} \frac{|F_k|!(|\mathcal{F}| - 1 - |F_k|)!}{|\mathcal{F}|!}(V(x_{i,F_k \cup f_j}) - V(x_{i,F_k})) \tag{2}$$

Going back to example setting 2 (Figure 1b), we see once either $P1$ or $P2$ enters a coalition that does not contain either of them (so $\{\varnothing\}$ or $\{P3\}$) the prediction increases from zero to one and will not increase further when the other enters. Because we average across all coalitions and in half of the coalitions, $P1$ will occur before $P2$ and in the other half $P2$ will occur before $P1$, the symmetry property will be satisfied and $P1$ and $P2$ will be given equal attribution. We can also see that if $P3$ does not change the prediction value in any coalition, it will be given zero attribution, satisfying the dummy property.

In the machine learning case, we would like to use a machine learning model M as the function that assigns a predicted value to x_i. Because a machine learning model requires a datapoint to have values for all features, we must approximate $V(x_{i,F_k})$ using other datapoints. Let x_l be a randomly sampled real datapoint from the dataset that is not x_i. We define a fake datapoint $x_{i,F_k,l}$ as a hybrid datapoint that contains the feature values of x_i for the features in F_k and the feature values of x_l for the features not in F_k. In our running example, $x_{i,F_k,l} = [ID\text{: } x_{l,ID}, T\text{: } 7, P1\text{:}6, P2\text{:}x_{l,P2}]$. $M(x_{i,F_k,l})$ is then used to approximate $V(x_{i,F_k})$.

Shapley values require summing over all possible coalitions and are very computationally expensive. There are algorithms that compute an approximate solution through sampling such as the method proposed by Vstrumbel et al [20]. In our case, we use an implementation, TreeSHAP [13, 14], designed to efficiently and quickly calculate exactly Shapley Values for decision tree based models.

Algorithm 1: Suggest Intervention for $Pair_{ij}$

```
Input : Dataset of Preexisting Log Files ($\mathcal{D}$), Set of
        Actionable Features ($\mathcal{F}_a$), Set of other
        features ($\mathcal{F}_o$), Mapping of Actionable
        Features to Interventions (GetIntervention),
        $student_i$ log file($L_i$) at n$^{th}$ attempt on
        $problem_j$, wheel-spinning Model Output
        Probability Threshold (p)
Output: Suggested Intervention for $Pair_{ij}$
          // We abbreviate wheel-spinning as WS
WSModel = TrainModel($\mathcal{D}$, $\{\mathcal{F}_a, \mathcal{F}_o\}$, n)
$X_i$ = GetCurrentFeatures($L_i$, $\{\mathcal{F}_a, \mathcal{F}_o\}$)
q = WSModel.predict($X_i$)
if q > p then
    $\{SHAP_a, SHAP_o\}$ = ComputeShapley($X_i$,
     WSModel, $\{\mathcal{F}_a, \mathcal{F}_o\}$)          // Section 3.1
    MaxFeature = $argmax_{f_a} SHAP_a$
    Intervention = GetIntervention(MaxFeature)
else
    Intervention = Don't Intervene
end
```

3.2 Baselines

We compare to two baselines and, in this section, include discussion for building intuition for which situations our prosed method could outperform the baselines.

Baseline 1 - Overall Feature Importance (FI): Because we are using a decision tree based method to predict wheel-spinning, overall feature importances are calculated automatically. Therefore, we can consider a method that when a student-item pair is predicted to wheel spin, choose the intervention suggested by the feature with the highest overall feature importance. This method requires less compute as it does not require an additional step of calculating individualized feature attributions. Conceptually, this method will perform equivalently as our proposed method when there is a single cause for wheel-spinning. However in cases where there can be many potential causes (for example, some student-item wheel-spinning is due to forgetting effects from long durations between learning while others are due to unmastered prerequisites), then this baseline, which will only select the single, most predictive cause for all students, will perform poorly. In this respect, this baseline has parallels to a baseline which predicts the majority class. Note that we do not compare to a baseline that predicts the majority class because we are considering a setting where we do not have any labels for wheel-spinning causes. Consequently our method has no way of discerning what the majority cause is. The goal of our work instead is investigating the effectiveness of feature attribution methods to identify causes.

Baseline 2 - Logistic Regression (LR): Linear models such as logistic regression are a computationally efficient subset of our method as they, by nature and without needing additional calculation, have feature credit attribution for the predictions of individual datapoints. They can potentially work well in cases where a linear relationship can accurately model the relation between features and wheel-spinning. However in many domains, such as CWTL, non-linear models for the wheel-spinning prediction can achieve better performance (shown in Section 5.4). Therefore in this work we focus on a method that can work with non-linear models and we treat linear models as a baseline.

4. SIMULATIONS

We assess the performance of our method in simulation where we can create true causes of wheel-spinning, which we define as needing 10 or more attempts on one educational activity to match both prior work [11, 12] and evidence from the CWTL data.

We simulate students using two different student models both based on the Bayesian Knowledge Tracing (BKT) model [10]. The BKT model is a two state Hidden Markov Model (HMM) and is a popular model of student learning that has been shown to be successful for various applications in the educational technology literature (for example Corbett et al. [10]). The model has two hidden states, mastered or not mastered, and two observed states, correct or incorrect. From the mastered state of a skill, the student will answer an educational activity involving that skill correctly unless they slip and answer incorrectly with a probability of slip (P(s)). From the unmastered state of that skill, a student will answer a problem involving the skill incorrectly, unless they guess correctly with a probability of guess (P(G)). Everytime the student is presented a practice opportunity for an unmastered skill, they have a probability of transitioning (P(T)) to the mastered state for the skill. We make modifications to the BKT model to match aspects of the CWTL domain that may also occur in other domains. In our simulations we specifically consider a situation where a student may have been moved on too fast because they passed a prerequisite by guessing. This is because the corresponding intervention of reviewing the relevant prerequisite could be automated and is a key feature we are trying to achieve in the CWTL setting.

4.1 Simulated Curriculum

In Figure 2a we illustrate our simulated sequence of educational activities as well as the prerequisite structure between them. In this setting we consider each activity as corresponding to a unique skill. Skills build on each other in the way shown in the prerequisite graph. To mimic the CWTL curriculum, simulated students are presented educational activities in order starting at A1. They are repeatedly presented an educational activity (for example A1) until they succeed and are moved onto the next activity (in our example, A2).

We note that while our analysis and results are in a setting where the curriculum is linear, our method does not rely on this setting and can be applied more generally to different types of ordering constraints over educational activities.

4.2 Student Model 1

In our first student model, we make two modifications to the BKT model to reflect behaviors that occur in our domain and in other domains. In CWTL, each activity involves answering a certain percentage of multiple choice questions relating to the target skill of the activity correctly. In this setting, the probability of guess starts low, however questions are reused between activity instances so the probability of guess increases with attempts as students may start to memorize answers. This effect can also occur in other domains where questions are reused. To mimic this effect in simulation, we start the probability of guess at a low base value P(G) and with every attempted answer by the student, we increase it in such a way that at the n^{th} attempt of the student on the problem the probability of guess, $P_n(G)$, is $P_n(G) = 1 - (1 - P(G))^n$. We use this function as it monotonically increases to its limit of 1.

Our second modification is, for skills involving prerequisites ($A4$ and $A5$), we enforce the prerequisite structure by defining a new transition probability for when the prerequisites are not mastered, $P_{unmastered}(T)$. In all our simulations this was set to zero however this probability can also be set to a small non-zero probability with similar results. This is to reflect the difficulty of learning complex combinatorial skills without mastering the prerequisites.

4.2.1 Data Generation

In our simulations, we show our method is able to correctly distinguish when and which prerequisite should be reviewed. We consider the whole population comprised equally of two different populations of students. Students of student population 1 finds all skills "easy" to master and has high transition probabilities for all skills. Students of student population 2 finds one of the prerequisite skills (A1, A2, or A3) "hard" to master and has low transition probabilities for that skill. For this student model, we are able to control which prerequisite students of student population 2 may not master by setting that prerequisite as "hard". Additionally we can examine the performance of our method at suggesting interventions in a heterogeneous population.

We report results from one set of parameters with the transition dynamics described in Table 1. Notice P(G) is lower for A4 and A5 to reflect the complexity of those two questions over A1, A2, and A3. We generate both our training and test sets by simulating 1000 student trajectories, 500 from each population. For these simulation parameters, initially $P_{easy}(T)$ is higher than $P_n(G)$ and if a student needs a low number of attempts on a prerequisite, they are most likely part of student population 1 and have mastered the prerequisite. If a student needs a higher number of attempts on a prerequisite, then they are most likely in student population 2 and they may either have mastered the prerequisite or passed through guessing and need to repractice the prerequisite. Decreasing the value of $P_{easy}(T)$ or $P(G)$ can increase the strength of this correlation between attempts and mastery and allow model accuracy to increase. Similarly, increasing these parameters, or increasing $P_{unmastered}(T)$ can decrease accuracy.

Table 1: Parameters for Student Model 1

P(T) "easy"	P(T) "hard"	P(G) (A1,A2,A3)	P(G) (A4,A5)	P(S)	"hard" skill
0.5	0.01	0.01	0.005	0	A2

Table 2: Parameters for Student Model 2

P(T)	P(D)	P(G) (A1,A2,A3)	P(G) (A4,A5)	P(S)
0.5	0.1	0.01	0.005	0

4.3 Student Model 2

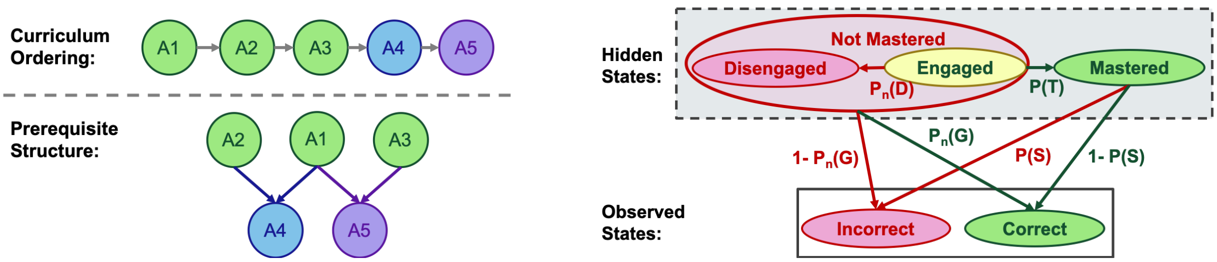
We designed our second simulated model to account for student engagement and simulate disengagement and wheel-spinning behavior. We did so based on expert insights, and findings from prior literature on boredom and disengagement in tutoring systems. A figure illustrating this modified model is shown in Figure 2b.

In this model we make an additional modification on Student Model 1 by splitting the "Not Mastered" state into two states: "Engaged" and "Disengaged". Each student for each activity starts in the Engaged state. In the Engaged state the student is open to learning and can transition to the "Mastered" state with probability P(T). However with each failed activity attempt, on the n^{th} attempt they can also transition to the "Disengaged" state with probability $P_n(D)$. This probability of disengagement starts at 0 and is parametrized by a base value of P(D). It increases monotonically in the same way the probability of guess does, to eventually reach 1: $P_n(D) = 1 - (1 - P(D))^{n-1}$. Once in the disengaged state for a skill, the student can transition out of it with probability $P(E)$. In our simiulations we set $P(E)$ to 0 however it can also be set to a small non-zero probability with similar results.

We make these modifications to reflect points from (1) prior literature and from domain expert insights that suggests repetitive tasks can lead to boredom [22, 8], (2) literature suggesting boredom can lead to disengagement which results in gaming behavior (such as random guessing) [3, 2, 4, 1, 11] as opposed to productive learning (3) literature suggesting disengagement and boredom are affective states that persist and are hard to transition out of [4, 1, 18].

4.3.1 Data Generation

We generate both our training and test set by simulating and generating 1000 student trajectories. Parameters used



(a) Simulated Curriculum

(b) Diagram of the modified BKT of Student Model 2

Figure 2: Simulated Student Setting

to generate the results are given in Table 2. For these parameters because $P(T)$ is initially much higher than $P_n(G)$ and $P_n(D)$, if a student needs a low number of attempts, they most likely mastered the activity. If they need a large number of attempts, they most likely became disengaged and guessed correctly. In these simulations, the correlation between attempts and mastery can be increased by increasing $P(T)$ or decreasing either $P(G)$ or $P(D)$. Similarly changing the parameters in the opposite direction or increasing $P_{unmastered}(T)$ or $P(E)$ can decrease accuracy.

4.4 Features

We train our model to predict wheel-spinning on the later skills, A4 and A5, and automatically suggest interventions in the form of if and which prerequisite to review. In both of the student models, needing a higher number of attempts on an activity is positively correlated with a skill not being learned. With this in mind we use the following three features and corresponding interventions: (1) Activity identity (A4 or A5): If assigned a high contribution, the corresponding intervention could be redesiging the level. (2): Number of attempts on the most recent prerequisite as defined by the prerequisite graph. The corresponding intervention would be to have the student review that activity. (3): Number of attempts on the second most recent prerequisite.

4.5 Results

4.5.1 Evaluation Metrics

To evaluate the accuracy of our method, we consider the frequency with which the method predicts a correct action, which includes correctly deciding to not intervene and correctly suggesting a correct intervention. We refer to student-problem pairs that would lead to wheel-spinning if no intervention is given as a wheel-spinning pair and student-problem pairs that would not wheel spin if no intervention is given as non-wheel-spinning pairs. Across all student-problem pairs, we define four counts:

1. Correct-Pairs_No-Intervention (CP_NI): the number of student-problems where the model correctly suggests no intervention)

2. Correct-Pairs _Intervention (CP_I): model correctly suggests the right intervention

Student Model	n	Method	Accuracy	Precision	Recall	F1	AUC
1	0	XGB LR	88% 86%	0.68 0.71	0.72 0.50	0.70 0.59	0.89 0.89
	5	XGB	94%	0.79	0.93	0.85	0.97
2	0	XGB LR	83% 80%	0.75 0.75	0.58 0.44	0.65 0.55	0.79 0.79
	5	XGB	93%	0.81	0.99	0.90	0.96

Table 3: Simulation wheel-spinning prediction results averaged over 200 simulations. XGB refers to XGBoost, LR refers to the Logistic Regression baseline. At n= 5 attempts the performance of XGB and LR are very similar so only the XGB results are included. At n = 0 attempts, XGB has higher Acuracy and F1 than LR.

3. Missed-Pairs (MP): model either suggests an incorrect intervention or incorrectly does not suggest an intervention)

4. Interrupted-Pairs (IP): model incorrectly suggests giving an intervention when it is unneeded, or suggests the wrong intervention

Note that IP and MP both include students that were wheel-spinning but the model suggests the wrong intervention since such students are both not helped ("missed") and would be asked to do something not useful ("interrupted"). Additionally, we classify wheel-spinning students who mastered both prerequisites and were still jumped back to a prerequisite as CP_I as insights from our domain expert suggests that jumping back when a student is wheel-spinning and possibly disengaged can be a helpful intervention.

Let S be the total number of student-problem pairs and define *accuracy* as the total percentage of student problem pairs that were given a correct intervention ($= \frac{CP\_NI+CP\_I}{S}$); *miss rate* as the percentage of wheel-spinning instances that were not identified or which were proposed the incorrect intervention ($= \frac{MP}{CP\_I+MS}$), and the *interrupted rate* as the percentage of Interrupted Pairs out of all student-problem pairs that did not need an intervention ($= \frac{IP}{CP\_NI+IP}$).

4.5.2 Results

For all results, we averaged over N=200 simulations by repeating 200 times the data generation procedure outlined in Sections 4.2.1 and 4.3.1. With this size of N, the standard deviation for all results reported in this section is less than 0.005 (for results reported in percentages, less than 0.5%).

Student Model	n	Method	Accuracy	Miss Rate	Interrupted Rate
1	0	Ours	88%	28%	8%
		LR	86%	50%	5%
	5	Ours	92%	14%	8%
		LR	92%	14%	8%
2	0	Ours	83%	42%	8%
		LR	80%	56%	5%
		FI	75%	68%	16%
	5	Ours	92%	4%	10%
		LR	86%	25%	17%
		FI	84%	34%	19%

Table 4: Simulation intervention suggestion results, averaged over 200 simulations. Ours refer to our proposed method, LR refers to the Logistic Regression baseline, FI refers to the overall XGBoost feature importance baseline. Notice the FI baseline was not included for Student Model 1 because in that simulation, there was only one cause of wheel-spinning (Prerequisite 2) so FI is exactly equivalent to our method.

We report the results of the XGBoost and Logistic Regression (baseline) models for predicting wheel-spinning in Table 3. For lower values of n, XGBoost can achieve higher accuracy and F1 when predicting wheel-spinning. As n increases, the dataset becomes heavily skewed towards datapoints with wheel-spinning as well as students needing less than n attempts correctly automatically labelled as no-wheel-spinning, resulting in both methods achieving high accuracy.

We report the results of our method for identifying interventions for both student models in Table 4 when making the prediction at 0 attempts and 5 attempts ($n = 0$ and $n = 5$). The probability threshold of the wheel-spinning model over which we suggest an intervention (p) was set to 0.5 for both. Our approach achieves high accuracy for both student models even when making early predictions before the student begins an activity (0th attempt). Additionally our method is mostly able to do better than the Logistic Regression baseline (LR). For Student Model 1, because there is only one cause of wheel-spinning the prescriptions of the XGBoost Overall Feature Importance Baseline (FI) was exactly the same as our method. However in Student Model 2 where there is more than one cause of wheel-spinning, our method performs much better.

Due to the fact that students are modelled stochastically, we are not able to achieve 100% accuracy as the correlation between number of attempts on a problem and problem mastery is not perfect. However we can increase the accuracy by making the prediction at a later number of attempts as shown in Table 4 when the intervention prediction made at the fifth attempt ($n = 5$). Our accuracy for both student models increases and the miss rate for both decreases. As we increase the attempt number at which we consider providing an intervention, all the student problem pairs that resulted in less than 5 attempts were correctly not intervened upon and automatically categorized as CP_NI. We provide further discussion of this hyparameter and the p hyperparameter in the Discussion (Section 6).

5. CAN'T WAIT TO LEARN

Our method was motivated by our collaboration with the Can't Wait to Learn (CWTL) program of War Child Holland. CWTL is a tablet based, curriculum aligned, self-paced, autonomous learning program that aims to teach basic numeracy and literacy skills to children in conflict-affected settings who are facing challenges in accessing quality education. The program is delivered on a tablet and targets learning objectives from grade 1-3. Based on the context, the program can be used as a standalone or a supplemental educational program. CWTL is currently rolled out in Sudan, Lebanon, Jordan, Chad, Bangladesh and Uganda. Prior studies found the program was able to result in increased psychological well-being as well as positive learning outcomes in multiple countries [7, 21].

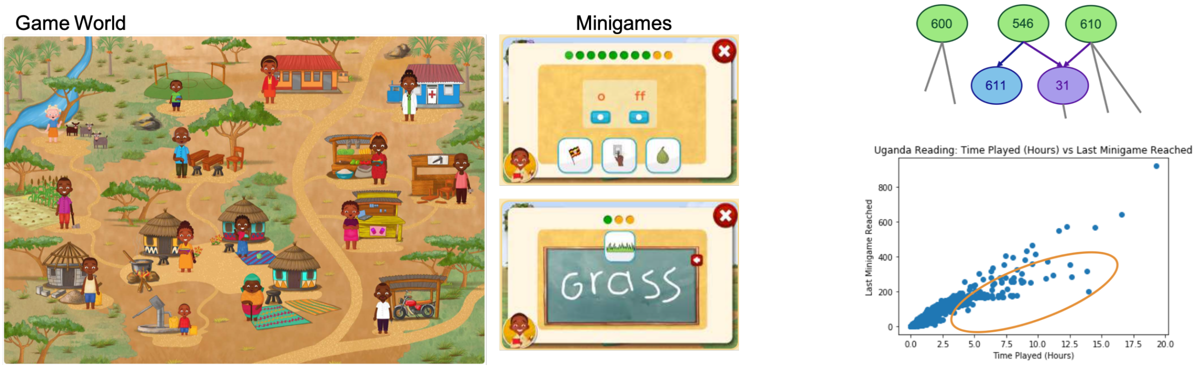
5.1 Game Mechanics

For our application we focus on the English reading program in Uganda where we notice a high amount of wheel-spinning. In classrooms utilizing the program, the instructor to student ratio is large, with class sizes of 114 students per teacher on average. The game takes place in the game world shown in the left panel of Figure 3a. In the game, the student is a member of a Ugandan village and the overarching narrative of the game is to help each village member achieve their goals by playing educational mini-games. The educational mini-games (Figure 3a right panels give two examples) and the instructional videos explaining concepts, such as letters or more complex vowel sounds, form the main educational mechanism. Each educational activity in the program is a specific instance of a mini-game and the curriculum is a fixed linear curriculum of a sequence of these educational activities. For example, in the mini-game at the top right of Figure 3a, the goal concept is learning to combine sounds of words beginning with "o". In the specific practice question shown of this mini-game, students first tap the blue buttons to listen to the sounds the "o" and "ff" components of the word make separately. To answer the question correctly, they must then tap the correct picture describing the complete word ("off"). To succeed on the activity students must answer 8 out of 10 instances of this question correctly as described by the green the orange circles displayed at the top. Students practice each activity repeatedly until they achieve this success criteria. When a student succeeds at an educational activity they are progressed to the next activity in the curriculum.

5.2 Wheel-Spinning Details

In analyzing the data, we find that 2.4% of student-problem pairs exhibit wheel-spinning. This is lower than in other systems because there exist easier activities for entertainment, engagement, morale, and for gaining initial familiarity with a new concept without too much cognitive overload. Wheel-spinning is still a problem as we find that 51% of students wheel spin at least once. The bottom plot of Figure 3b shows time played compared with the last activity reached. The students who are below the curve whom we would like to help are circled in orange.

To determine the threshold of attempts to define wheel-spinning, we examine plots of student attempts on the game they are currently playing at the end of the most recent log file. If students are stuck on an activity, they are spending more time on it and have a higher probability of being



(a) The Educational environment

(b) Top: Part of the prerequisite structure between minigames, Bottom: wheel-spinning

Figure 3

on that activity when the playing session ends. Therefore the activity the student is currently playing at the time the log file was accessed is correlated with activities students are wheel-spinning on. We compared the distribution of attempts of the problem students are currently on to the distribution of attempts on the activities they played 1 or 5 activities ago, which are less correlated with wheel-spinning. We find a non-negligible percentage of students need 10 or more attempts on the current activity they are playing (27%) while few students require 10 or more attempts on activities played 1 activities (6.8%) or 5 activities (5.9%) ago. We therefore defined wheel-spinning as failing 10 or more attempts on an activity.

5.3 Model

In our model we used the following actionable features and describe the corresponding intervention. We highlight the actionable features that allow for in game interventions in **bold**. These are especially helpful in our domain where student to teacher ratios may be large. We also provide an example of a non-actionable feature[1]:

(1) **Last Played**: If there has been a long duration since the student last played, the intervention is to diagnose and have them review what they forgotten. (2) **Number of attempts on the Prerequisite 1, 2 and 3 Prerequisites ago**: A small portion of the prerequisite structure is shown in the top image of Figure 3b. These features use the prerequisite graph to find the last, second to last, and third to last prerequisite in the curriculum. These features in the CWTL domain can be evidence that a student did not master the corresponding prerequisite. The intervention is to have the student practice the prerequisite. (3) Mini-game Type: Allows the model to identify if a mini-game should be redesigned. (4) Number of attempts on the first video: To pass any video, a student only needs to watch it completely. The number of attempts on the first video can be an indicator of low technological fluency. The intervention is to have a notification that encourages them to ask a teacher or a peer for help. (5) First Time Mini-game Type Seen?: Students generally will need more attempts the first time they experience a mini-game. So this feature, while not actionable, allows the model to make more accurate predictions.

5.4 Results

We first examine the accuracy of our model at predicting wheel-spinning. We used data from 1170 students. Students were assigned randomly to the training and test set with 80%, or 943, students assigned to the training set. The students completed 60 activities on average. There were a total of 55,035 student-activity pair datapoints in the training set with 1,294 of them as wheel-spinning (2.4%). There were a total of 15,004 datapoints in the test set with 322 of them as wheel-spinning (2.2%). These datapoints were all used in the $n = 0$ condition. Considering only the student-activity pairs that required 5 or more attempts ($n = 5$), the training set had 2568 datapoints (50% wheel-spinning - there were still 1,294 wheel-spinning datapoints since only datapoints with less than 5 attempts were removed) and the test set had 664 datapoint (48% wheel-spinning). At $n = 9$, the training set had 1454 datapoints (89% wheel-spinning) and the test set had 365 datapoint (88% wheel-spinning).

As shown in Table 5, while our accuracy is quite high, due to the class imbalance, precision, recall, and F1 are low. We tried a variety of different models such as CART decision trees and Random Forests and we found the model we used, XGBoost, to do the best by a slight margin over Random Forests and significantly over CART. We additionally report results for Logistic regression to show that for lower values of n, it is not able to achieve the same accuracy as XGBoost. As with the simulations, as the value of n increases, the accuracy difference of the two models on predicting wheel-spinning decreases as both models achieve high accuracy at higher values of n. This is due both to a higher balance of wheel-spinning datapoints in the dataset and automatically correctly predicting not-wheel-spinning on students who needed less than n attempts. However this increased accuracy at higher values of n is at the expense of allowing some of the students who will eventually wheel-spin

[1] We also included other non-actionable features to reduce confounding and improve prediction accuracy which we omit in sake of clarity and brevity. Some examples of other non-actionable features included were the number of times minigame was seen before, the Learning Level, which gives a rough location of where the student is in the curriculum, as well as other features helpful for distinguishing current student location in curriculum.

n	Method	Accuracy	Precision	Recall	F1	AUC
0	XGB	93%	0.21	0.60	0.31	0.91
	LR	88%	0.12	0.60	0.19	0.86
5	XGB	98%	0.60	0.60	0.60	0.99
	LR	98%	0.53	0.58	0.55	0.99
9	XGB	99.7%	0.90	1	0.94	0.999
	LR	99.7%	0.88	1	0.94	0.99

Table 5: Wheel-Spinning Prediction: XGB refers to XGBoost, LR refers to the logisitc regression baseline. LR has worse predictive accuracy and a lower F1 score than XGB when the prediction is made at lower values of n.

on a problem still spend multiple attempts on the problem. To deploy a system we would work with a domain expert to decide the n that would be best.

To verify the method, we compare our method's predictions to those an expert would prescribe. To obtain the expert prescription, we blinded the domain expert author of this paper, by showing them the cases and asking for their prescriptions before sharing with them the details or results of the model. To generate the test cases, we randomly sampled true wheel-spinning student-problem pairs of that were also predicted as wheel-spinning by the model. To get diverse cases, sampling was done by throwing out newly sampled cases that were very similar to two or more previously selected cases, until we had 6 cases total. For purposes of making a comparison, we made a list of possible causes and interventions for the domain expert to choose from, including a none-of-the above choice. In our model, some features allow for immediate actions (reviewing a prerequisite problem) while others do not (redesigning an educational activity). The immediately actionable features are much easier to intervene on and based on our expertise gained, are much more favorable to an expert or instructor. To reflect this, we made the decision (before discussing the methods and giving the examples to the expert) to choose the maximum immediately intervenable feature if its Shapley value is greater than half of the maximum feature Shapley value.

The cases are shown in Table 6. The expert's prescription and the suggestions of various algorithms are shown in Table 7. Overall we found that our method can be promising for automatically suggesting correct interventions. Our method's suggested interventions agreed with the domain expert's prescribed interventions 4 out of 6 times, but not in Cases 1 and 6. Additionally our method performed better than logistic regression and the highest overall XGBoost feature importance baselines.

Features	Case1	Case2	Case3	Case 4	Case 5	Case 6
Mini-game (MG)	31	31.	611.	31	31.	546
Last Played (s) (LP)	34.	10.	6	10	12	25
First Time Seen? (F?)	F	F	F	F	F	T
Attempts Prereq1 (P1)	1	1.	1	7	12	∅
Attempts Prereq2 (P2)	1.	1.	1.	1	1.	∅
Attempts Prereq3 (P3)	1.	4.	1.	3	4.	∅

Table 6: The 6 cases from the CWTL dataset used for qualitative evaluation of the methods.

	Expert	Ours	LR	FI
Case1	∅	P3	P3	P3
Case2	**P3**	P3	P3	P3
Case3	**MG**	MG	MG	P3
Case4	**P1**	P1	P3	P3
Case5	**P1**	P1	P3	P3
Case6	**P1**	F?	F?	P3
Accuracy	-	4/6	2/6	1/6

Table 7: A comparison of our method and various baselines with the Expert's prescription. Ours refers to the method described in this work, LR refers to the logistic regression baseline, and FI refers to the XGBoost overall feature importance baseline. MG refers to the "Mini-game" feature, F? refers to "First Time Seen?" featuree, P1, P2, and P3, refer to "Attempts Prerequisite1", "Prerequisite2" and "Prerequisite3" respectively and ∅ refers to an expert prescription not in the list of what the model can suggest.

In Case1, the expert believed the exact identity of the educational activity, a feature we did not include was the true cause of the wheel-spinning and the intervention would be to redesign that particular activity. While we did include the mini-game type of each activity, we did not include the unique identity of each activity in the model as it would result in too many features compared to the amount of data we had. Therefore one tradeoff of our method that needs to be made when there is limited data is using as many features as we can to catch all possible causes and using only the most important subset of the features to maintain model robustness. In Case 6, even though the prerequisite structure was created together with the domain expert, during the activity of prescribing interventions, the expert realized there may have been an incorrect dependency in the graph. Where under the original graph there were no prerequisites for this activity, under the new prerequisite graph this activity would have prerequisites. This case highlights the importance of having the correct curriculum graph.

In both the incorrect cases it would not have been feasible for our method to have obtained the correct answer, suggesting the ability of our method to identify correct interventions given correct inputs.

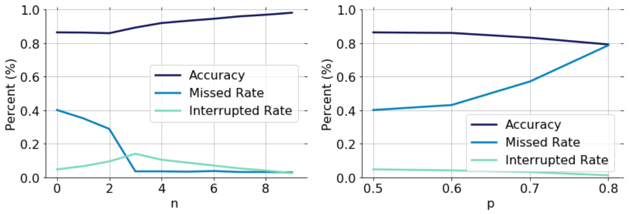
6. DISCUSSION

6.1 Possible Improvements With More Data

The program is currently running and data is being collected. As the amount of data increases and even more expressive function classes, such as neural networks, can be robustly trained, it is possible for the model to become more accurate. Additionally currently we have limited data, especially of the wheel-spinning class, therefore we do not include all possible helpful features, such as exact activity identity, to ensure model robustness. This omission can cause errors such as in Case 1. As more data becomes available this tradeoff between including features and model robustness becomes less important. More features can be included for more accurate intervention predictions.

Table 8: Parameters for Student Model 1

P(T) "easy"	P(T) "hard"	P(G) (A1,A2,A3)	P(G) (A4,A5)	P(S)	"hard" Skill
0.5	0.01	0.01	0.005	0	A2



(a) Sweeping n ($p = 0.5$) with Student Model 2

(b) Sweeping p ($n = 0$) with Student Model 2

Figure 4: Sweeping Hyperparameters

6.2 Setting Hyperparameters

As shown in both the results sections, our prediction at the 0th attempt of a student activity pair (before the student starts an activity) can be inaccurate. As we increase the number of attempts, n, before we intervene, we are able to increase accuracy as we by default do not intervene on students who need less than n attempts. However this increased accuracy comes at the expense of letting the students who will wheel spin spend time unproductively attempting the activity. This tradeoff may also not be feasible in environments where students may dropout before n attempts such as educational games played in a casual setting. We illustrate the miss rate decreasing and the accuracy increasing as we increase the number of attempts on which we make the prediction for Student Model 2 (Section 4.3) in Figure 4b. We fix the threshold probability of the wheel-spinning model output to make prediction (p) at 0.5.

Another key design choice touched upon is setting p, the threshold of the wheel-spinning model output for classifying wheel-spinning. To give a concrete example, changing the threshold from the default 0.5 to 0.7 would mean we need the wheel-spinning model to output a probability of 0.7 on a student-activity pair before we decide to suggest an intervention. Therefore at every attempt, we can trade off between correctly suggesting an intervention for a student-question pair and "interrupting" students by changing the certainty threshold. We examine this tradeoff using simulations following Student Model 2 (Section 4.3) at $n = 0$ and plot this in Figure 4b. As expected, as we increase the threshold, the missed rate increases as the interrupted rate decreases.

6.3 Limitation: Does Not Establish Causality

One limitation of this method is causal inferences cannot be made. To illustrate this we consider simulations following the simulation procedure of Student Model 1 (Section 4.2) under a new set of parameters given in Table 8. In this case we make A2 difficult instead of A1. As shown in Figure 2a, A2 only affects A4. Students who struggle due to unmastered prerequisite skills only struggle on A4. There will be very few students who, due to randomness, will struggle on A5. Therefore A4 will be positively correlated with wheel-spinning. However the design of A4 is not the direct cause of most students' struggling where the true cause is the lack of mastery on A2. Looking into the Shapley values, A4 is chosen incorrectly as the highest valued feature for 11% of all true positive wheel-spinning cases. This can inaccurately lead to an assumption that A4 needs to be redesigned. While redesiging A4 could indeed reduce the number of students wheel-spinning on A4, if students master A2, they will not struggle more on A4 than they would on A5. Therefore suggesting reviewing A2 instead of redesigning A4 as the most likely intervention candidate would be desired as reviewing A2 is often a much lower overhead intervention than redesigning A4. Coming up with solutions for this issue would be an interesting direction of future work.

7. CONCLUSIONS

In this work we propose a method to automatically suggest interventions for wheel-spinning students. To our knowledge this is one of the first investigations of both designing a wheel-spinning model to suggest immediately actionable interventions as well as using interpretable machine learning methods such as Shapley values in educational technology. We evaluate our method's ability to suggest useful interventions by investigating the correctness of the suggested intervention in two different simulations and through a qualitative investigation comparing the interventions suggested by our method and the interventions prescribed by the expert. We found our method had high accuracy and was able to choose an accurate intervention for more than 80% of the time in the simulations before the students begin an activity. Additionally in our real world setting our suggestions mostly agreed with the expert prescription and the other cases were due to limitations of the model and errors made in the inputs to the model. Our results suggest our method can help inform interventions and improve educational systems to be more effective and engaging.

8. ACKNOWLEDGEMENTS

We would like to thank all our collaborators at War Child Holland and the Can't Wait To Learn team. This research was supported by the NSF CAREER award, NSF big data award and the GFSD Fellowship.

9. REFERENCES

[1] J. M. L. Andres and M. M. T. Rodrigo. The incidence and persistence of affective states while playing newton's playground. In *7th IEEE international conference on humanoid, nanotechnology, information technology, communication and control, environment, and management*, 2014.

[2] R. Baker, J. Walonoski, N. Heffernan, I. Roll, A. Corbett, and K. Koedinger. Why students engage in "gaming the system" behavior in interactive learning environments. *Journal of Interactive Learning Research*, 19(2).

[3] R. S. Baker, A. T. Corbett, K. R. Koedinger, and A. Z. Wagner. Off-task behavior in the cognitive tutor classroom: when students" game the system". In *Proceedings of the SIGCHI conference on Human factors in computing systems.*

[4] R. S. Baker, S. K. D'Mello, M. M. T. Rodrigo, and A. C. Graesser. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, 68(4).

[5] J. Beck and M. M. T. Rodrigo. Understanding wheel spinning in the context of affective factors. In

International conference on intelligent tutoring systems.

[6] J. E. Beck and Y. Gong. Wheel-spinning: Students who fail to master a skill. In *International conference on artificial intelligence in education.*

[7] F. Brown, A. Farag, F. Hussein, L. Miller, K. Radford, A. Abdullatif Abbadi, K. Neijenhuijs, H. Stubbe-Alberts, T. de Hoop, J. Turner, A. Jetten, and M. Jordans. Can't wait to learn: A quasi-experimental mixed-methods evaluation of a digital game-based learning programme for out of school children in sudan. *Under Review in the Journal of Development Effectiveness.*

[8] G. Chanel, C. Rebetez, M. Bétrancourt, and T. Pun. Emotion assessment from physiological signals for adaptation of game difficulty. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(6):1052–1063, 2011.

[9] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.*

[10] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4).

[11] Y. Gong and J. E. Beck. Towards detecting wheel-spinning: Future failure in mastery learning. In *Proceedings of the second (2015) ACM conference on learning@ scale.*

[12] S. Kai, M. V. Almeda, R. S. Baker, C. Heffernan, and N. Heffernan. Decision tree modeling of wheel-spinning and productive persistence in skill builders. *JEDM| Journal of Educational Data Mining*, 10(1).

[13] S. M. Lundberg, G. G. Erion, and S.-I. Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[14] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems.*

[15] S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K.-W. Low, S.-F. Newman, J. Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering*, 2(10).

[16] N. Matsuda, S. Chandrasekaran, and J. C. Stamper. How quickly can wheel spinning be detected? In *EDM.*

[17] R. Narayanam and Y. Narahari. A shapley value-based approach to discover influential nodes in social networks. *IEEE Transactions on Automation Science and Engineering*, 8(1).

[18] M. M. T. Rodrigo. Dynamics of student cognitive-affective transitions during a mathematics game. *Simulation & Gaming*, 42(1).

[19] L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[20] E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3).

[21] J. Turner, K. Taha, N. Ibrahim, K. I. Neijenhuijs, E. Hallak, K. Radford, H. Stubbé-Alberts, T. de Hoop, M. J. Jordans, and F. L. Brown. A mixed-methods evaluation of an innovative, digital game-based learning programme to improve educational outcomes of out-of-school children in lebanon. *In Submission to the Journal of Education in Emergencies.*

[22] J. J. Vogel-Walcutt, L. Fiorella, T. Carper, and S. Schatz. The definition, assessment, and mitigation of state boredom within educational settings: A comprehensive review. *Educational Psychology Review*, 24(1).

[23] H. Wan and J. B. Beck. Considering the influence of prerequisite performance on wheel spinning. *International Conference on Educational Data Mining*, 2015.

[24] C. Zhang, Y. Huang, J. Wang, D. Lu, W. Fang, J. Stamper, S. Fancsali, K. Holstein, and V. Aleven. Early detection of wheel spinning: Comparison across tutors, models, features, and operationalizations. *International Conference on Educational Data Mining*, 2019.

[25] Y.-J. Zhang, A.-D. Wang, and Y.-B. Da. Regional allocation of carbon emission quotas in china: Evidence from the shapley value method. *Energy Policy*, 74.