How Long is Enough? Predicting Student Outcomes with Same-Day Gameplay Data in an Educational Math Game

Rachel Harred
North Carolina State University
rlharred@ncsu.edu

Christa Cody
North Carolina State University
cncody@ncsu.edu

Mehak Maniktala
North Carolina State University
mmanikt@ncsu.edu

Preya Shabrina
North Carolina State University
pshabri@ncsu.edu

Tiffany Barnes
North Carolina State University
tmbarnes@ncsu.edu

Collin Lynch
North Carolina State University
cflynch@ncsu.edu

ABSTRACT

Curriculum-integrated games can provide teachers with data to help them decide when and how to intervene with individual students. Based on our prior work observing teachers using ST Math, teachers may not be able to attend to a dashboard or student screens to determine who might need intervention. We therefore set out to determine how much data we need from the current ST Math gameplay session to predict performance. Based on the available log data that tracks student performance over SETS of puzzles, we performed two experiments to predict performance. The first uses data from one game level, which is about 3 minutes long, to predict the performance on the next level, and the second uses the first 6 minutes of gameplay to predict how many levels a student can complete in 20 minutes, a typical class length. Our results show that our data are not fine-grained enough to allow for paired level prediction, but that 6 minutes of gameplay can be used to rank students in order of performance for a class session. These results can be used as a basis for an alert system that could help teachers prioritize their time in the classroom.

1. INTRODUCTION

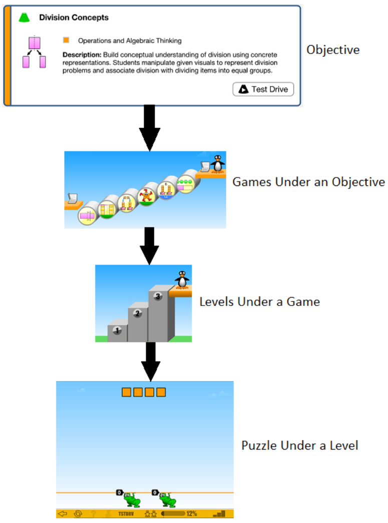
Educational games can be a useful tool for teachers to provide additional practical learning for students [3]. As more educational games become curriculum-integrated, a significant portion of a students time can be spent in these systems. However, teachers cannot monitor and assist each student at the same time, struggling to identify students who need help the most. In previous work, we observed teachers assistance often was influenced by things such as classroom layout and disruptive behavior rather than learner proficiency or needs [13]. Furthermore, the work identified that students who “struggled quietly” often went unnoticed. In other work, the authors found that when students possibly need intervention but do not receive it, they might get frustrated and give up or replay an easier game instead [9]. Other research has also shown that teachers can often unintentionally favor or give assistance to certain types of students due to differences in perceptions or help-seeking behaviors [4, 22, 5]. Therefore, providing teachers with information to help them determine who needs assistance the most may be crucial to some low-performing students.

Despite the amount of data gathered with each playthrough, teachers in our system are only provided with a student’s current progress in the curriculum and a feature that allows a student to “raise” their hand through the system. However, this is only visible on the student’s screen via a purple hand indicator and often goes unnoticed. Therefore, we sought to determine if there was a way to provide teachers with knowledge regarding students projected progress as fast as possible, so that the teachers can determine who to help from there. With the machine learning techniques that can process such data and help predict outcomes, we wanted to find the correct technique to answer our question. Machine learning and educational data mining techniques have been successfully used in educational game research for many years [11, 21, 10, 18].

In this paper, we tried to determine the smallest amount of time needed to predict student outcomes for one gameplay session by investigating multiple feature selection algorithms and prediction models on student gameplay data for an educational game, Spatial Temporal Math (ST Math). We tried two methods of prediction using data analysis and machine learning: 1) Trying to predict student outcomes for playing one level of a game using gameplay data from only the previous level, 2) Using the least amount of time of a student’s gameplay data to predict the number of levels they will pass in the next twenty minutes of gameplay. To accomplish this, we tried various machine learning and feature selection methods to find the most significant features needed to predict student outcomes in this educational game. In this study our intention was to give insight to the teachers of ST Math by indicating our best guess for which students could most benefit from teacher intervention on a single day, provided early enough in the gameplay session to allow the teacher to help as many as possible.

1.1 Spatial Temporal Math (ST Math)

ST Math is a curriculum-integrated supplemental mathematics game for 2nd-4th-grade students that uses spatial puzzles to teach basic math concepts [19, 12, 14, 15]. The puzzles do not contain any textual instruction. The games are grouped at the highest level by objective which indicates the broad math concept. Each objective contains a number of games, the gameplay under an objective varies but concerns the same content inside an objective. The games usually have between 3 to 5 levels each, and the gameplay across levels is similar but increases in difficulty. There are usually between 6 and 8 puzzles per level. The puzzles are either randomly generated using a template or randomly selected from pre-designed puzzles depending on the level. Each puzzle requires the student to do the correct action to indicate their answer. Animated feedback is presented to the student following the puzzle solving attempt that shows the student if they are correct or incorrect. For example, in the game "Fair Sharing" under "Division Concepts" a student is asked to distribute boxes equally among animals to construct a straight bridge and will show the bridge blocked off or with gaps that make it impossible to cross in case of an incorrect answer. A level begins with a set number of lives, usually 2, that resets at the beginning of each level. If a student's response is incorrect, the student loses a life. If the student loses all of their lives before completing the level, they do not pass the level and must retry it. To pass a level, the student must complete all puzzles without losing all their lives. After a student passes a level, they may move on to the next level in the game or objective, or backtrack and play a previously passed level. We refer to this backtracking as replay. A level attempt includes passing a level, failing a level, and replay. Each student has the option to ”raise their hand” if they want help from their teacher by clicking on a hand icon on the screen. The teacher also has access see which objectives and levels a student has passed. See Figure 1 for a breakdown of ST Math.



Description: Build conceptual understanding of division using concrete representations. Students manipulate given visuals to represent division problems and associate division with dividing items into equal groups.

Figure 1: ST Math

2. RELATED WORK

Educational games in classrooms are helpful to teachers because the students can receive individualized attention and learning from the game while the teacher gives one-on-one attention to students who need it.[7]. However, teachers have limited time and try to prioritize their attention to the students who need it most. It has been shown that students who are given attention by teachers have increased student learning[5]; therefore, teachers who are able to focus their attention to low-performing students should see them benefit.

Unfortunately, there are many reasons that students who need help do not receive it. One study found that middle-class students seek help more directly than working-class students and end up getting more help as a result[4]. In a recent study of classroom observations using ST Math, researchers found differences in classroom format have an influence on who receives help[13]. Furthermore, this work found teachers in free-seating classrooms could not easily see the raised hand indicator on student screens and so those help-seeking students went unnoticed. In classrooms that use rotation-seating, teacher attention was only given if the ST Math group was being disruptive. In general, the students who directly asked for help or were obviously off-task received more teacher intervention than the students who were less vocal [13].

The task of automatically identifying students who need help has been explored [2]. With machine learning methods that can process large amounts of data and make predictions, many have been using these methods in attempt to solve the problem. Ahadi et al. explored machine learning techniques and were able to use the first week of data in a programming course to predict student performance with accuracy ranging from 71-80%. Additionally, Jiang et al. employed logistic regression models to predict the type of certificate a learner received in Massive Open Online Courses (MOOCs) [8]. In a study at the Open University, decision tree models were implemented on users' current and previous activity to predict if they were at risk of failing a module [24]. Another Open University study explored using Bayesian models to build real-time predictive models from student data and found little difference among the types of models but that the accuracy increased with the addition of data throughout the progression of the learning module [23]. These studies show that machine learning can be used to predict and understand student behavior, but are not being used to directly aid in student learning.

Predictive models are now being integrated into teacher dashboards or alert systems to enable this aid. In a survey of K-12 teachers who used intelligent tutoring systems in class, Holstein et al. found that they were very interested in having real-time classroom monitoring tools that would help them decide which students most needed their attention [7]. In another study, Holstein et al. developed a teacher alert system using smart glasses that showed real-time indicators of student behaviors floating above their heads [6]. Their early findings suggest that this helped direct the teachers to the students who needed intervention the most [7].

3. DATA

The data is collected by MIND Research Institute, who created ST Math. This study was conducted on data from 3rd grade students who played ST Math during the 2016-2017 school year. The data contains 31 objectives, 154 games, and 669 levels which equals 5,186,269 total level attempts by 8,983 students from 111 schools and 636 teachers. We excluded students who completed objectives not contained in the 3rd grade objectives which removed 21,544 level attempts. These are students who might have been erroneously included from other grades, as the system is used for grades 3-5. For the purposes of our study, we filtered out level attempts where the student only completed one level attempt in our 30 minute gameplay session cutoff. This removed 101,849 level attempts, leaving us with a final dataset of 5,062,876 unique level attempts. The initial data we were provided with had 6 features for each level attempt: STMathID (unique ID for each student), Level, Objective Code, Timestamp, Number of Correct Puzzles, and Total Number of Puzzles. For the purpose of our analysis, we created additional features shown in 1.

4. METHOD

We wanted to explore different ways of providing teachers with information about student progress, so that they could intervene and monitor progress according to their own classroom goals. Therefore, our intention is to predict the projected progress of a student using the least amount of information and using only same-day gameplay data. Here, we are comparing two methods of data segmentation: predicting the time spent passing the next level based on the time spent passing the current level, and using the least amount of gameplay to predict how many levels would be passed in the next 20 minutes of gameplay. We are attempting to see if a student's performance in the beginning of a gameplay session is a good prediction for their later performance. A gameplay session is defined as subsequent level attempts that are separated by less than 30 minutes. If two level attempts are separated by 30 minutes or longer, we count the second attempt as a new gameplay session. We decided on a 30 minute cutoff because a pause in the game of 30 minutes or more might indicate the student was working on something else in between and we cannot say that the previously played level will have any effect on the performance of next level. However, we still want to include students who may be truly struggling, receiving help from a teacher or giving help to another student during the playthrough of a level. Only 3.5% of the data for the gap between play sessions was between 30 minutes and 20 hours, while 11.6% of the data was 20 hours or longer between gameplay sessions.

5. EXPERIMENT 1: PAIRWISE PREDICTION

This section details our attempt at using previous level data to predict student outcomes for the next level attempt.

5.1 Pairwise Method

We wanted to see if we can predict how well a student will do on the next level by using only the data from the previous level in the prediction. Our aim in this experiment was to investigate if the features of a single level for a student can predict whether the student would have needed an intervention for the next level. This would be beneficial to teachers because it would give an alert immediately after a student finishes a level that would tell them that the student might need help on the next level. We used pairwise prediction, so the data was grouped into *Level A* and *Level B*, with the constraint that *Level A* and *Level B* had to be in the same objective and the same gameplay session. We considered *Level A* to be the first attempt on a new level, all subsequent attempts (retries) until the level was passed, and any replayed levels that happened before or after the level was passed. Any gameplay that happened between the first attempts of two consecutive levels would be counted as *Level A* data. *Level B* is the next attempt on a new level after *Level A*, and contains the same information as *Level A*: number of attempts, retries, replays before and after passing, up until the next new level attempt. We expected the number of attempts to pass a level to provide information about how many attempts they will need for the next level because the levels increase in difficulty inside objectives and this is

Table 1: Features Created for the Analysis

Feature	Description
game	The name of the game this level is in, extrapolated from the Objective Code and Level
performance	Number of puzzles correct before losing all lives divided by total puzzles in level
isReplay	1 if this level has been attempted before, otherwise 0
isUnneceReplay	1 if the level is replay, otherwise 0
prevPassAttempt SameLevel	For replay: 1 if it is on the same level played in previous row, 0 if not, NA if not replay
passedCurrentLevel B4Unnece	1 if the student passed the level in the previous row (their "current" level) AND this row is replay, 0 if not, NA if not replay
sameObj	1 if current level and previous are the same objective
levelPlayTimeSec	Timestamp - previous row's timestamp
gameplaySession	Numbered starting at 1 and incrementing if the levelPlayTimeSec > 30 minutes (1800s). Rows with the same numbered gameplaySession are assumed to happen during a single "play session"
prevPerf	Average of previous gameplay session's performance
firstInGameplaySession	1 if this level attempt is the first in a new gameplay session, otherwise 0
lastInGameplaySession	1 if this level attempt is the last in a gameplay session, otherwise 0

consistent through ST Math. Also, research has shown that replay that happens before passing a level results in a negative effect on performance while replay that happens after passing a level has been shown to have a positive effect[12]; therefore, we expected this data to also be useful for the prediction.

Due to time constraints and the complexity of the feature creation, we used a subset of our total dataset for this analysis. The dataset includes 830 students, and 665 unique objective-level pairs. Objective-level pairs are level pairs within an objective. However, some students did not complete all the objective-levels resulting in a total of 277,975 unique student objective-level pairs.

5.1.1 Pairwise Feature Generation

Since the raw data only included gameplay aspects per attempt such as time taken, attempt performance and the kind of attempt (retries, replay, etc.), we engineered 33 additional features for every student objective-level pair. There are 7 attempt categories and 5 metrics per each category. The 7 attempt categories are as follows: overall level attempts, total retry and replay attempts, retry attempts, total replay attempts, total replay attempts before passing the current level, total replay attempts after passing, replay attempts of the same level (current) after passing, and replay attempts of other levels after passing. The 5 metrics for each category are as follows: whether an attempt category occurred (except for overall attempts category), total number of attempts (except for overall attempts category), total time, average time, and average performance.

Next, for each student, we identified the consecutive levels (*Level A, Level B*) within each objective that were completed in the same session. We found a total of 222,258 such level pairs for 830 students. We explored four different ways to define an intervention: if the total time was greater than the 75th percentile (I-TotalTime), if the average time was greater than the 75th percentile (I-AvgTime), if the average performance was less than 25th percentile (I-AvgPerf), and if the student could not finish a level in the first attempt (I-FirstAttempt). Each of these intervention types were intended to capture a different aspect of a student's ability to complete a level.

5.1.2 Pairwise Feature Selection and Prediction Models

The analysis was carried out in Python. We normalized the time-related features and then explored three feature selection techniques in the scikit-learn[16] package. We used a pipeline of a feature selection wrapper method, SelectFromModel, with models such as LinearSVC (Linear Support Vector Classifier with L1 loss), LassoCV (Lasso linear model with 3-fold Cross Validation), and Logistic Regression. We used 7 classifiers KNN (n = 3), LinearSVC (Linear Support Vector Classifier), Decision Tree (using Gini index), Random Forest (using Gini index), MLP (Multi-layer Perceptron classifier with a 2-layer (100,100) neural network using a learning rate of 0.001 and reLU activation function), ADAboost, Naive Bayes and measured the prediction accuracy using 10-fold Cross Validation.

Table 2: Features Selected using Linear SVC (with L1 loss) for Pairwise Prediction

Feature	Description	Mean (mode for Binary)	SD/ occurrence for Binary
retryOrReplayBinary	1 if retryAndReplaySum>0, otherwise 0	0	0:181,804 1:40,454
replayBinary	1 if replaySum>0, otherwise 0	0	0:221,981 1:277
retryAndReplayTimeSum	Total Time Spent on Retries and Replays	65.78	200.18
avgTimePerLevel	Average Overall time	225.49	175.23
avgPerfTotal	Average Overall Performance	0.94	0.16
avgRetryAndReplayPerf	Average Replay Performance	0.16	0.35
avgRetryPerf	Average Retry Performance	0.16	0.35

5.2 Pairwise Results & Discussion

The feature selection based on LinearSVC with L1 loss and the Random Forest classifier provided the most optimal prediction accuracy. Table 3 shows the results of using a Random Forest classifier for each intervention type. We observed that the average time spent on *Level A* was selected for each intervention type based target for *Level B*. The distribution parameters of the features selected are shown in Table 2. We observed that very few features related to the replays were selected. This may be because the consecutive level pair dataset recorded very few rows with retries or replays (18.20%) and even lower just considering replays (0.12%). Such a high degree of sparsity in replay made any replay related features not significant enough to contribute towards the predictions. Another interesting observation is that the performance over all the attempts (avgPerfTotal) in *Level A* was not a significant predictor of the intervention for *Level B* in any of the models primarily because of the small variance recorded for this feature. The small variance is due to the granularity of the data only recording passed level attempts with failed puzzles as 100% performance. On the other hand, the average time in *Level A* (avgTimePerLevel) was a significant predictor of the intervention for *Level B* for every intervention type based target. We recorded few replays in general and a low variance in the performance related features, so only the time related features were varied enough to capture the relationship between *Level A* and *Level B*. The results suggest that the average time spent on an attempt in a level is the most significant predictor of whether a student may need assistance in the next level. However, the classifier models for each intervention type did not perform significantly better than a baseline classifier that would predict all the observations to be the Majority Class (the class

Table 3: Features Selected based on Linear SVC with L1 loss and Prediction Accuracy with Significant Predictors using a Random Forest classifier for Each Intervention Type Target

	I-TotalTime	I-AvgTime	I-AvgPerf	I-FirstAttempt
Feature Selected	avgTimePerLevel avgPerfTotal retryOrReplayBinary retryAndReplayTimeSum avgRetryPerf	avgTimePerLevel avgPerfTotal avgRetryAndReplayPerf replayBinary	avgTimePerLevel avgPerfTotal retryOrReplayBinary	avgTimePerLevel
Significant Predictor	avgTimePerLevel: 0.7 retryAndReplayTimeSum: 0.17	avgTimePerLevel: 0.93	avgTimePerLevel: 0.98	avgTimePerLevel: 1.00
Majority Class	76.64%	76.18%	99.59%	75.32%
Prediction Accuracy (K=10)	77.20%	77.21%	99.60%	76.49%

containing more students) as shown in Table 3. This suggests that the relation between the behavior of students in two consecutive levels may be highly varied and that it is difficult to generalize whether an intervention is needed in a level based on only one previous level. It may also suggest that such a prediction may be dependant on how far along students are in their academic year. To investigate the first scenario, we added a feature for a student's previous performance average, an average of every level attempt until now, in attempt to help distinguish low-performing students from the rest. Previous performance average was selected for each intervention type prediction but had lower feature importance (0.03%) because of the low variance and, therefore, did not affect the prediction accuracy. To investigate if the time of the academic year had any impact on the prediction, we added a feature to indicate the month in which the sessions occurred. Similar to the previous performance average, this feature was selected but had a low feature importance (0.05%) leading to an insignificant difference in the prediction accuracy.

Since only time related features were varied enough to capture the variance in the student behavior in two consecutive level pairs, we explored ways other than feature generation to perform the pairwise prediction. We sliced the data based on aspects, such as replay type or month of the year but, again, obtained similar prediction accuracies; however, the replay related features did get selected and had high importance for the prediction in the data sliced by replay type. To investigate if the variance in the content of the objectives may be affecting the prediction results, we performed the prediction for each intervention type within a single objective and observed that the prediction accuracy decreased slightly. This suggests that even within one objective, the behavior of a student in one level, as captured in its current granularity, may not accurately predict if they need an intervention in the next level.

The pairwise prediction models may not have generated desirable results because we may need more than just one previous level's data to predict if an intervention is needed. There is not sufficient data about each level in this dataset to accurately represent the student's performance and to create good predictions. Therefore, having more fine-grained details about level attempts, including knowing more about how the levels compare to each other within each objective, may improve the prediction accuracy.

6. EXPERIMENT 2: LEVELS COMPLETED IN 20 MINUTES

We chose to determine if we could predict the number of levels completed in 20 minutes using only information from the current session. Using only information from the current session will allow an easier integration with the current system with minimal changes needed. Different schools and classrooms have unique ways of using ST Math[13]. As a result, there is a variety of session times ranging from very short (less than 5 min) to sessions lasting over an hour, with an average of 23 minutes spent in a session. Therefore, we decided to predict how many levels a student would complete in a 20 minute session. This information could be used by teachers to identify students who will not be able to complete the number of levels the teacher expects for that session and the teacher can intervene to assist or encourage. With this prediction, the system could provide a teacher with each student's predictions and order the students by the lowest predicted number of levels to complete in the next 20 minutes. Then, the teacher can easily look at the slowest students and make the judgment, based on their knowledge of each student and what goals the teacher has for that lesson, and determine who they need to assist. Studies have shown that teachers may focus assistance on students with better help-seeking behaviors because they are often more persistent or better in requesting help[4, 20]. Providing this information this early in the session could be crucial for low-performing student with who are not asking for help or not doing so effectively.

6.1 Levels Completed Prediction: Methods

The data we used consisted of 787949 session observations from 8978 unique students, 111 schools, and 636 teachers. A session represents a period of time that the student spends working on ST Math without taking longer than a 30 minute break (see Section 4 for full definition). For accurate predictions, we chose to use the first 6 minutes of gameplay due to the average level attempt taking approximately 3 minutes. We refer to this segment of data used for prediction as a time “slice”. Since our goal was to use the least amount of information to do the prediction, we wanted this time to be as short as possible. We initially attempted to use shorter time slices, but due to a level attempt taking on average 3 minutes, this did not provide a sufficient amount of data to represent the students’ gameplay behavior and, in some cases, eliminated slower students data for that time slice.

Next, we removed sessions under 10 minutes (242750 obs. - 169863 obs. under 6 min) and sessions over 75 minutes (10,257 obs.). We chose these cutoffs to eliminate short sessions where predictions would not be useful and long sessions, in some cases over 4 hours, that were likely anomalous.

Table 5 shows the statistics for session and slice features. The average session time was 28 minutes and the average slice length was 4.5 minutes.

Table 4: Session and Time Slice Stats for the time, performance, and levels completed

	Feature	Mean(SD)	Mdn
Session	Levels per 20 min	3.8 (2.5)	3.5
	Total Time (min)	28.9(26.3)	816.2
	Avg Performance	74.9%(23.4)	81.3%
Slice	Levels Completed	1.9(1.1)	1
	Total Time (min)	4.5(0.9)	4.6
	Avg Performance	72.6%(34.0)	100

6.1.1 Levels Completed Prediction: Feature Generation

From the level data, the data was segmented into the first 6 minute time slice for prediction. Features were aggregated from this 6 minute time slice to capture what each student was able to do, such as complete a level, fail a level, retry a level, or engage in replay. The features generated are based on performance, time, level attempts/replay features, the objective the student was in within that slice. The **performance features** were: average performance (numeric), percentage of levels passed out of all attempts (numeric), and percentage of levels completed out of all attempts (numeric). The **time features** from the slice were: the total time (numeric), the average level time (numeric), number of passed levels per time (numeric, scaled), number of completed levels per time (numeric, scaled), and the month of the session. The **level attempt features** were: total number of replays (numeric), total number of levels failed (numeric), total number of levels passed (numeric), total number of levels completed (numeric), total puzzles attempted (numeric), total puzzles completed (numeric), whether they engaged in replay (binary), whether they re-attempted a level (binary). Then, there were **31 binary features representing the objective** the student was playing when the session started.

6.1.2 Levels Completed Prediction: Model Selection and Feature Selection

For this prediction, we tried a variety of models, tuning of these models, and alteration of the target variable. Model and feature selection was accomplished by using scikit-learn[16]. For the models, we tried both classification and regression. For classifications, multiple groupings of number of completed levels in 20 minutes were chosen using balanced classes, the best accuracy (77% using a 2-layer neural network) being a split to determine if a student could complete at least an average number of levels in 20 minutes. However, we decided that regression, providing finer-grained predictions, would provide more useful information to the teachers and allow them to have more autonomy in deciding which students need help. For regression, we tried to predict how many levels a student could complete in 20 minutes, which derived from taking the total number of levels completed and the total time of the session and scaling.

We tried multiple models, including decision trees, neural networks, and random forests after normalizing the features. Intrepretable machine learning methods are more important because knowing which features are more influential to predicting performance can give insights to how students learn in games. In the results, we show the 2 best models compared to a baseline. The baseline model is created by always predicting the mean of the completed levels per 20 minutes. The best models were created by testing multiple models and fine-tuning the parameters. The two best models are created from scikit-learn: a 3-layer (50,30,20) neural network (MLPRegressor) using a learning rate of 0.001 and reLU activation function, and a Random Forest (RandomForestRegressor) using mean squared error as the criterion function and setting the minimum samples for a split to be 20.

To evaluate each model, we used the following metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), adjusted R-squared score, and explained variance (EV). We choose these metrics to evaluate, on average, how accurate each prediction rate was to determine if the error is small enough to still provide a good estimate of the students’ projected progress. Both R^2 and EV were used to evaluate the variance of these errors and check for biases within the models. All models were evaluated with 10-fold cross validation.

We attempted feature selection using scikit-learn filter methods, such as feature importance from tree regression, and a wrapper method (SelectFromModel) with each model. Feature selection did not improve any models, and resulted in significantly worse predictions in most cases. This is most likely due to the limited amount of features available. Because our data is not fine-grained, we have a limited amount of information about each student for a level attempt. This indicates that each feature could be providing key information regarding their current progress. Therefore, for the best models the whole feature set was used (see Feature Generation).

6.2 Levels Completed Prediction: Results & Discussion

This section discusses the results of the Experiment 2 regressions.

Table 5 shows the results of the evaluation metrics for the two best regression models compared to the baseline model. The NN and the Random Forest both perform similarly, both outperforming the baseline model. Although the MAE does not have a large difference, the RMSE is much lower. This indicates that the variance of the errors is significantly smaller for our predictive models. The MAE of 1.2 for our predictive models means that on average the prediction will only be around 1 level off for a specific student, which still provides a good estimation for the teacher to use. The adjusted R-squared and explained variance are almost identical for both models, which happens when the mean of the errors is approaching zero. Although these scores are not perfect, in the context of educational data from a system used with multiple teaching styles, this is a highly meaningful result[1, 17].

Table 5: Results for the 2 best regression models compared to a baseline(mean).

Model	*MAE*	*RMSE*	*R2*	*EV*
Forest	1.24	1.59	0.58	0.58
NN	1.22	1.58	0.59	0.59
Baseline	1.96	2.48	-1.2E-5	0.0

Table 6 shows the top five most important features for decisions in the Random Forest. All of these top features focus on the number of completed levels, the total time, or a combination of these features. This is not surprising because the amount of levels a student can complete in the first 6 minutes should be a good indication of how they will perform over the whole session. However, this is assuming that the students remain seated and playing the game in the same manner.

Table 6: The top five most important features from the Random Forest

Rank	**Feature**
1	Total Levels Completed
2	% of Levels Completed
3	Completed Levels per Time
4	Total Time
5	Average Level Time

Figure 2 shows a density plot of the predicted values versus the actual values, the yellow/lightest color being the highest density. Both figures show the highest density areas occurs closely to the actual values. The Neural Network appears to have a higher density closer to the line and the points appear to be more compact, although both models show similar predictions. Both figures are zoomed in to focus on the lower level number predictions, although few points have values higher than 10. One note is both models are less fitted for the higher values and tends to predict around 10 after the actual value is 10+. However, we are mostly concerned with students who are completing very few levels. If a student falls into the 10+ range of levels completed, the actual value becomes less important due to how much above the average it is. A teacher will still be able to use this information to identify over-performing students and ensure they don't get too far ahead of the class.

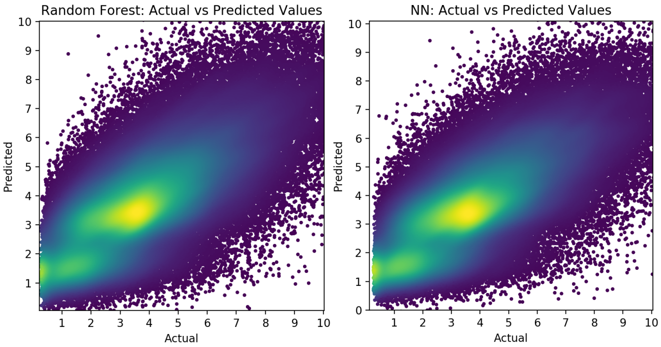
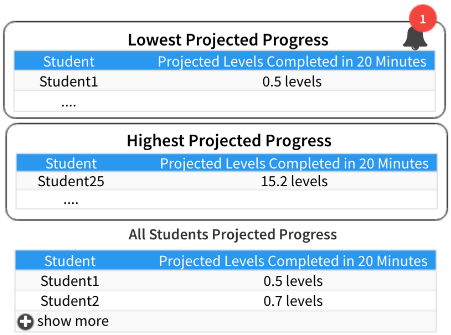


Figure 2: These density plots show the predicted vs. actual values for the two best regression models to predict the number of completed levels in 20 minutes. Note: the yellow/lightest areas represent the highest density of points.

With the variability of how this system is used, the models evaluations are a positive result. For example, during field observations of the system, we found many teachers asked students ahead in the curriculum to help students next to them during a gameplay session. Therefore, a student may spend part of the session working as normal, then, after the teacher has identified a struggling student, the teacher may ask the student next to them to help. This could result in much higher predicted values than what the student actually completes. Furthermore, we observed students in some classrooms initially talking and working at a slower pace in the first few minutes of a session as they settled in, then shortly being asked to focus. This could result in much lower predictions of the projected number of levels that student could complete. Since the data does not only include the sessions where students quietly work by themselves for a continuous period of time, accurate predictions are difficult.

Furthermore, the data we used focused on only the same-day gameplay data, not containing any information regarding how a student has previously performed in other sessions. This decision was made to limit the changes required to implement this system in the game. However, including prior information may improve predictions. One possible way to control for the effect of the different teaching styles is to include teacher or school information in the model. However, this would create very sparse features due to the large number of teachers and schools that use the system. A future attempt could identify and categorize the teachers or schools based on similar styles and add those features to the models.

This prediction can be used in two main ways: identifying the lowest performing students who may need assistance, but may not be requesting help, and identifying students who may be working too fast and getting ahead in the curriculum. The second usage may not seem like an issue, but having a large knowledge gap between students may make a classroom harder to manage and teach. This is a problem teachers seek to avoid in ST Math that they have remedied by asking those students to help others or by allowing them to play games while others catch up[13]. For ease of use, these predictions could be provided in a simple list with each student's name and the predicted number of levels they will complete in 20 minutes. Furthermore, the top 5 lowest and highest predictions could be presented at the top of the interface so teachers could quickly have an idea of who is struggling and who may need to be slowed down. Because teachers already have access to where each student is in the curriculum, the teacher can use their expertise and knowledge of the students to make judgement calls on what to do from there. A mock interface of how this could be presented can be seen in Figure 3.



Lowest Projected Progress

Student	Projected Levels Completed in 20 Minutes
Student1	0.5 levels
....	

Highest Projected Progress

Student	Projected Levels Completed in 20 Minutes
Student25	15.2 levels
....	

All Students Projected Progress

Student	Projected Levels Completed in 20 Minutes
Student1	0.5 levels
Student2	0.7 levels
⊕ show more	

Figure 3: Mock interface showing how teachers would view students' predictions

7. OVERALL DISCUSSION

The results for the levels completed experiment were more promising than the pairwise experiment. For the pairwise prediction, the lack of fine-grained puzzle level data made it difficult to predict whether a student may need intervention based only on their previous level's data. We believe the results for this method of pairwise prediction might improve with more data about how the objectives, games, and levels relate to each other. On the other hand, the prediction model from the levels completed experiment had decent results with the MAE and RMSE indicating that the predictions are generally within 1-2 levels of the actual completed levels for the 20-minute time period. Having additional information, including finer-grained puzzle-level data, should also improve this prediction.

Providing the teachers with a projected completed amount of levels allows us to give the teachers a list of the students ranked by the number of levels they are predicted to complete. This allows the teachers to use their expertise to distinguish the higher- and lower-performing students during that game session, and, importantly, the teachers have the ability to make judgments about interventions according to their discretion. Currently, the teachers only have information on student progress in the overall game curriculum (which objectives each student has finished and how many levels have been completed). Additionally, the only method currently used to support students in seeking help is the raised hand indicator, which has been shown to not always get the teachers' attention due to its location on the students' screens. We believe that incorporating this prediction into the system will be a valuable tool for teachers that will suggest which students are struggling and allow them to decide if they need intervention. Giving teachers these suggestions after only 6 minutes of gameplay time means that the teachers will have more control over the classroom progress because they will have more time to help students get back on track instead of being behind for the entire session and be able to slow down students who are getting too far ahead of the class.

7.1 Limitations

To reduce the amount of time processing the data, we used a representative subset for the pairwise prediction. However, we compared multiple numerical and categorical features between this subset and the entire dataset and determined that it contained almost identical distributions of data points. We created histograms for the distributions of performance, level play time, levels in session, time of session, performance session, and compared the number of schools and teachers represented in the subset to the totals. We were only missing 6 out of 111 schools and we had students from almost half of the teachers (291 out of 636) included in our subset.

We do not have fine-grained interaction data, which means we cannot tell exactly how many puzzles a student gets wrong. This lack of information causes our data to be skewed by having many performance scores of 100%, without capturing the full gameplay. However, there are other features that we can use tease out this information, like level time, as students who pass a level while also getting puzzles wrong will most likely take longer because they are doing more problems. We have finer-grained puzzle level data, but it does not match up accurately with our level data. This means that while we can do studies on these datasets separately, we cannot combine them to have the full picture of what a student is doing during the level: which puzzles they see, if any puzzles are repeated during a level, how many puzzles right and wrong, and the time spent on each individual puzzle in a level. These finer granularities could offer valuable information on what a student is doing during a level and their performance compared to the whole student set.

8. CONCLUSION

This study aimed to use the least amount of student gameplay data possible to predict which students would benefit from teacher intervention during the remainder of the gameplay session. We tried two granularities of prediction for our analysis. We hypothesized that we could use one level's data (average of 3.5 minutes of gameplay) to predict the next level's outcomes, as this controls for content and difficulty, but this hypothesis was not confirmed. The lack of fine-grained level attempt data might not allow us to make a good prediction. Our second hypothesis was that we could use the first 6 minutes of gameplay (about 2 levels) to predict how many levels the student could complete in the next 20 minutes. This had a reasonable outcome with a MAE of 1.2 and RMSE error of 1.6, meaning that, on average, the prediction is only off by 1-2 levels, which is a good estimation of how many levels a student will complete. We believe this can provide a valuable resource for the teachers who use ST Math in their classrooms, to help them concentrate their time and energy on the students who need it the most. Furthermore, this method allows the teachers to have a certain level of judgment in regards to who needs the assistance, which is imperative in a system that is used in multiple styles. Future work could investigate how this affected the students' performance if we gave this information to teachers.

9. ACKNOWLEDGMENTS

This research is made possible by support of the National Science Foundation under Grant No. 1544273.

10. ADDITIONAL AUTHORS

Additional authors: Teomara Rutherford (North Carolina State University, email: taruther@ncsu.edu).

11. REFERENCES

[1] Abelson, R.P.: A variance explanation paradox: when a little is a lot. Psychological bulletin **97**(1), 129 (1985)

[2] Ahadi, A., Lister, R., Haapala, H., Vihavainen, A.: Exploring machine learning methods to automatically identify students in need of assistance. In: Proceedings of the eleventh annual International Conference on International Computing Education Research. pp. 121–130. ACM (2015)

[3] Backlund, P., Hendrix, M.: Educational games-are they worth the effort? a literature survey of the effectiveness of serious games. In: 2013 5th international conference on games and virtual worlds for serious applications (VS-GAMES). pp. 1–8. IEEE (2013)

[4] Calarco, J.M.: "I need help!" Social class and children's help-seeking in elementary school. American Sociological Review **76**(6), 862–882 (2011)

[5] Good, T.L.: Which pupils do teachers call on? The Elementary School Journal **70**(4), 190–198 (1970)

[6] Holstein, K., Hong, G., Tegene, M., McLaren, B.M., Aleven, V.: The classroom as a dashboard: co-designing wearable cognitive augmentation for k-12 teachers. In: Proceedings of the 8th International Conference on Learning Analytics and Knowledge. pp. 79–88. ACM (2018)

[7] Holstein, K., McLaren, B.M., Aleven, V.: Intelligent tutors as teachers' aides: exploring teacher needs for real-time analytics in blended classrooms. In: Proceedings of the seventh international learning analytics & knowledge conference. pp. 257–266. ACM (2017)

[8] Jiang, S., Williams, A., Schenke, K., Warschauer, M., O'dowd, D.: Predicting mooc performance with week 1 behavior. In: Educational data mining 2014 (2014)

[9] Karumbaiah, S., Baker, R.S., Shute, V.: Predicting quitting in students playing a learning game. In: EDM (2018)

[10] Lee, S.J., Liu, Y.E., Popovic, Z.: Learning individual behavior in an educational game: a data-driven approach. In: Educational Data Mining 2014 (2014)

[11] Liu, Y.E., Mandel, T., Butler, E., Andersen, E., O'Rourke, E., Brunskill, E., Popovic, Z.: Predicting player moves in an educational game: A hybrid approach. In: EDM. pp. 106–113. Citeseer (2013)

[12] Liu, Z., Cody, C., Barnes, T., Lynch, C., Rutherford, T.: The antecedents of and associations with elective replay in an educational game: Is replay worth it? In: EDM (2017)

[13] Peddycord-Liu, Z., Cateté, V., Vandenberg, J., Barnes, T., Lynch, C.F., Rutherford, T.: A field study of teachers using a curriculum-integrated digital game. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. p. 428. ACM (2019)

[14] Peddycord-Liu, Z., Cody, C., Kessler, S., Barnes, T., Lynch, C.F., Rutherford, T.: Using serious game analytics to inform digital curricular sequencing: What math objective should students play next? In: Proceedings of the Annual Symposium on Computer-Human Interaction in Play. pp. 195–204. ACM (2017)

[15] Peddycord-Liu, Z., Harred, R., Karamarkovich, S., Barnes, T., Lynch, C., Rutherford, T.: Learning curve analysis in a large-scale, drill-and-practice serious math game: Where is learning support needed? In: International Conference on Artificial Intelligence in Education. pp. 436–449. Springer (2018)

[16] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research **12**(Oct), 2825–2830 (2011)

[17] Prentice, D.A., Miller, D.T.: When small effects are impressive. Psychological bulletin **112**(1), 160 (1992)

[18] Romero, C., Ventura, S.: Educational data mining: a review of the state of the art. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **40**(6), 601–618 (2010)

[19] Rutherford, T., Farkas, G., Duncan, G., Burchinal, M., Kibrick, M., Graham, J., Richland, L., Tran, N., Schneider, S., Duran, L., et al.: A randomized trial of an elementary school mathematics software intervention: Spatial-temporal math. Journal of Research on Educational Effectiveness **7**(4), 358–383 (2014)

[20] Ryan, A.M., Gheen, M.H., Midgley, C.: Why do some students avoid asking for help? an examination of the interplay among students' academic efficacy, teachers' social–emotional role, and the classroom goal structure. Journal of educational psychology **90**(3), 528 (1998)

[21] Sabourin, J.L., Shores, L.R., Mott, B.W., Lester, J.C.: Understanding and predicting student self-regulated learning strategies in game-based learning environments. International Journal of Artificial Intelligence in Education **23**(1-4), 94–114 (2013)

[22] Skinner, E.A., Belmont, M.J.: Motivation in the classroom: Reciprocal effects of teacher behavior and student engagement across the school year. Journal of educational psychology **85**(4), 571 (1993)

[23] Wolff, A., Zdrahal, Z., Herrmannova, D., Kuzilek, J., Hlosta, M.: Developing predictive models for early detection of at-risk students on distance learning modules. In: Machine Learning and Learning Analytics Workshop at The 4th International Conference on Learning Analytics and Knowledge (LAK14). p. 24–28 (2014)

[24] Wolff, A., Zdrahal, Z., Nikolov, A., Pantucek, M.: Improving retention: predicting at-risk students by analysing clicking behaviour in a virtual learning environment. In: Proceedings of the third international conference on learning analytics and knowledge. pp. 145–149. ACM (2013)