2019-04-18

Characterizing Productive Perseverance Using Sensor-Free Detectors of Student Knowledge, Behavior, and Affect

Anthony Botelho
Worcester Polytechnic Institute

Characterizing Productive Perseverance Using Sensor-Free Detectors of Student Knowledge, Behavior, and Affect

by

Anthony F. Botelho

A Dissertation

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Learning Sciences & Technologies

May 2019

APPROVED:

Neil T. Heffernan, Ph.D., Major Advisor

Erin R. Ottmar, Ph.D.

Joseph E. Beck, Ph.D.

Ryan S. Baker, Ph.D., University of Pennsylvania

Abstract

Failure is a necessary step in the process of learning. For this reason, there has been a myriad of research dedicated to the study of student perseverance in the presence of failure, leading to several commonly-cited theories and frameworks to characterize productive and unproductive representations of the construct of persistence. While researchers are in agreement that it is important for students to persist when struggling to learn new material, there can be both positive and negative aspects of persistence. What is it, then, that separates productive from unproductive persistence? The purpose of this work is to address this question through the development, extension, and study of data-driven models of student affect, behavior, and knowledge. The increased adoption of computer-based learning platforms in real classrooms has led to unique opportunities to study student learning at both fine levels of granularity and longitudinally at scale. Prior work has leveraged machine learning methods, existing learning theory, and previous education research to explore various aspects of student learning. These include the development of sensor-free detectors that utilize only the student interaction data collected through such learning platforms. Building off of the considerable amount of prior research, this work employs state-of-the-art machine learning methods in conjunction with the large scale granular data collected by computer-based learning platforms in alignment with three goals. First, this work focuses on the development of student models that study learning through the use of advancements in student modeling and deep learning methodologies. Second, this dissertation explores the development of tools that incorporate such models to support teachers in taking action in real classrooms to promote productive approaches to learning. Finally, this work aims to complete

the loop in utilizing these detector models to better understand the underlying constructs that are being measured through their application and their connection to productive perseverance and commonly-observed learning outcomes.

Acknowledgements

I would like to acknowledge and thank the love of my life, Paige, for her continued and unwavering support and assistance and without whom I would not be where I am today. I further extend gratitude to my family - Paul, Donna, Joseph, Elizabeth, Wayne, Esther, and Tara - for their love and support through this period of my life.

As I could not hope to list all the ways in which I am grateful for the insights, support, and assistance that my friends and fellow lab-mates, both past and present, have provided to me over the years, I wish to extend my sincerest thanks to those below:

Luke	Dan	Neil
Megan	Avery	Erin
Lindsey	Taylyn	Joe
Meaghan	Seth	Ryan
Terrasa	Korinn	Jake
David	Doug	Ivon
Matthew	Siyuan	Cristina
Ilir	March	Tricia
Lia	Ashvini	JA
Kyle	John	Emma
Danielle	Hannah	Cindy
Shawn	Chris	Katie
Stephanie	David	Jenny

And while I stop where I have balanced columns, I thank everyone else who has supported and helped me over the years.

Funding

The work presented in the following pages has been supported by a number of funding avenues. Specifically, my work has been funded by a Graduate Assistance in Areas of National Need (GAANN) Fellowship from the U.S. Department of Education (P200A120238), and the Office of Naval Research (N00014-13-C-0127). More broadly, funding is acknowledged from NSF (grant #'s DRL-1316736, DRL-1252297, DRL-1109483, DRL-1031398, 0742503, ACI-1440753, DGE-1535428, 0231773, 0448319, 1822830), ONR's "STEM Grand Challenges," IES (grant #'s R305A120125, R305C100024, R305K03140, R305A07440), The Spencer Foundation, The Bill and Melinda Gates Foundation.

Contents

1 Introduction **1**

I Developing, Improving, and Applying Student Models to Study Learning **5**

2 The Prediction of Student First Response Using Prerequisite Skills **6**

2.1 Introduction 7

2.2 Background 9

2.3 Dataset 11

2.3.1 Methodology 12

2.4 Results 16

2.4.1 Comparison of Overall Performance 17

2.4.2 Comparison Over Individual Skills 19

2.5 Contribution 21

2.6 Conclusion and Future Works 23

3 Improving Sensor-Free Affect Detection Using Deep Learning **25**

3.1 Introduction 26

3.2 Dataset 29

3.2.1 Data Collection and Feature Distillation 29

3.3 Methodology . . . 30
3.3.1 Network Structure . . . 31
3.3.2 Handling Time Series Data and Labels . . . 32
3.3.3 Model Training . . . 34
3.4 Measures . . . 35
3.5 Results . . . 35
3.5.1 Adjusting the Dropout Context . . . 35
3.5.2 Comparing RNN Variants . . . 36
3.6 Discussion and Future Work . . . 38

4 Developing Early Detectors of Student Attrition and Wheel Spinning Using Deep Learning 41

4.1 Introduction . . . 42
4.2 BACKGROUND . . . 46
4.2.1 Wheel Spinning . . . 46
4.2.2 Student Attrition and Stopout . . . 47
4.2.3 Deep Learning in Educational Contexts . . . 48
4.3 Dataset . . . 50
4.3.1 Features . . . 52
4.3.2 Wheel Spinning and Stopout Labels . . . 52
4.4 Methodology . . . 54
4.4.1 Building Models of Wheel Spinning and Stopout . . . 55
4.4.2 Transfer Learning . . . 57
4.5 Results . . . 60
4.5.1 Metrics . . . 60
4.5.2 Model Performance . . . 62
4.5.3 Observing Model Performance by Opportunity . . . 67

4.6 Future Work 71
4.7 Contributions and Conclusions 72

5 Machine-Learned or Expert-Engineered Features? Exploring Feature Engineering Methods in Detectors of Student Behavior and Affect 75

5.1 Introduction 76
5.1.1 Expert-Engineered vs. Automatically Distilled Features 77
5.1.2 Research Questions 79
5.2 Background 80
5.3 Data and Labels 82
5.4 Methodology 84
5.4.1 Utilizing Expert-Engineered Features 85
5.4.2 Deep Learning Models 88
5.4.3 Machine-Learned Expert-Inspired Features 91
5.4.4 Exploring the use of Co-Training 92
5.5 Results and Discussion 94
5.6 Limitations and Future Work 96
5.7 Conclusions 98

II Using Detectors of Student Knowledge, Behavior, and Affect to Drive Action 99

6 The ASSISTments TestBed: Opportunities and Challenges of Experimentation in Online Learning Platforms 100

6.1 Introduction 101
6.2 The ASSISTments Ecosystem 102

6.2.1 The ASSISTments Testbed 103
6.3 Video vs. Text Feedback: A Case Study of RCEs within ASSISTments 105
6.3.1 Methods to Reduce the Standard Errors of Effects 109
6.4 conclusions 115

7 Putting Teachers in the Driver's Seat: Using Machine Learning to Personalize Interactions with Students 117

7.1 The Problem 118
7.2 The Opportunity 121
7.2.1 Results of Prior NSF Support: Heffernan's ASSISTments and other CoPIs 121
7.2.2 An Opportunity to Apply Google's Smart Reply Technique to the Education Problem 125
7.2.3 ASSISTments currently has detectors that rely on student input 127
7.3 The Solution 132
7.3.1 DRIVER-SEAT Vignette: What we want this to look like . . 133
7.3.2 Project Activities 136
7.3.3 Details of the system behind DRIVER-SEAT 140
7.3.4 Experimentation and Exploration 144
7.3.5 Research Questions Addressing Issues in Human Learning . . 145
7.3.6 Research Questions Addressing Issues in Computer Science . . 149
7.4 Broader Impacts 152

8 The HAND-RAISE Intervention through LIVE-CHART: Directing Teachers' Attention to Prevent the Loss of Student Interest in STEM 153

8.1 The Problem 153

8.2 The Opportunity 155
8.2.1 Results of Prior NSF Support: Heffernan's ASSISTments and other CoPIs 156
8.2.2 Opportunities through AI-Enhanced Classrooms 159
8.2.3 ASSISTments currently has detectors that rely on student input 160
8.3 The Solution 161
8.3.1 The HAND-RAISE Intervention Vignette 162
8.3.2 Project Activities 167
8.3.3 Method of Evaluation 173
8.4 Broader Impacts 179
8.5 Intellectual merit 179

III Understanding the Role of Student Knowledge, Behavior, and Affect in Productive Perseverance 180

9 Studying Affect Dynamics and Chronometry Using Sensor-Free Detectors 181

9.1 Introduction 182
9.2 Previous Work 184
9.2.1 Detectors of Student Affect 187
9.3 Methodology 188
9.3.1 Dataset 188
9.3.2 Affect Dynamics 190
9.3.3 Affective Chronometry 192
9.4 Results 193
9.4.1 Observing Affect Dynamics 193

9.4.2 Observing Affective Chronometry 194
9.5 Discussion and Future Work 197

10 Refusing to Try: Characterizing Early Stopout on Student Assignments 203

10.1 Introduction 204
10.1.1 Student Refusal 206
10.2 Background 208
10.3 Dataset 210
10.4 Methodology 213
10.4.1 Characterizing Early Stopout and Refusal 213
10.4.2 Categorizing Stopout Behavior 216
10.4.3 Stopout Behavior by Opportunity 218
10.4.4 Observing Student Confidence 219
10.5 Results and Discussion 220
10.6 Contributions and Future Work 225

11 Identifying the Constructs Underlying Models of Student Knowledge, Behavior, and Affect 228

11.1 Introduction 228
11.1.1 Given a variety of commonly used assessment measures of student success, what is the dimensionality of the constructs measured by these assessments? 229
11.1.2 What is the dimensionality of constructs measured by the observed detectors of student knowledge, behavior and affect? . . 230

11.1.3 What is the relationship between the learning constructs measured by the observed assessment measures and those constructs represented by the detectors of student knowledge, behavior, and affect? 230
11.1.4 Which constructs represented by the detectors of student knowledge, behavior, and affect are reliable predictors of short- and long-term outcomes? 231
11.2 Detectors and Outcome Measures 231
11.2.1 Detectors of Student Knowledge 231
11.2.2 Detectors of Student Behavior 233
11.2.3 Detectors of Student Affect 234
11.2.4 Measures of Unproductive Perseverance 236
11.3 Methodology 239
11.3.1 Data 239
11.3.2 Detector Models 242
11.4 Factor Analyses 245
11.5 The Relationship Between Factors 249
11.6 Predictive Models of Unproductive Perseverance and Performance . . 251
11.6.1 Observing Distal Student Performance 257
11.7 Conclusions 258

List of Figures

2.1 The hypothetical students and data shown, fabricated to show our methodology, exemplifies the table creation process. Using a training set, a probability table is created for each skill by categorizing students with similar performance history 13

2.2 The difference of RMSE per skill when comparing our method of binning to standard knowledge tracing, ordered from highest to lowest difference. The number above each skill indicates the p-value of the difference. 20

2.3 The difference of RMSE per skill when comparing our method of binning to majority class predictions, ordered from highest to lowest difference. The number above each skill indicates the p-value of the difference. 21

4.1 A simplified representation of the LSTM model structure, illustrating how information flows from previous timesteps to inform each model estimate. 54

4.2 A visual example of the transfer learning procedure. The hidden layer of the trained LSTM model is used as input to train each a decision tree and logistic regression to predict each wheel spinning and stopout behavior. 55

4.3 The performance of the LSTM model in predicting within-assignment wheel spinning by opportunity. 65
4.4 The performance of the LSTM model in predicting next assignment wheel spinning by opportunity. 65
4.5 The performance of the LSTM model in predicting within-assignment stopout by opportunity. 69
4.6 The performance of the LSTM model in predicting next assignment stopout by opportunity. 70

6.1 An example experimental design with ASSISTments comparing text-based feedback with video-based feedback when students request help. 106
6.2 The estimated treatment effects of student completion for each of the 9 experiments across all methods. 113
6.3 The estimated treatment effects of student completion for each of the 9 experiments across all methods. 114

9.1 The proposed theoretical model of affect dynamics as presented by D'Mello and Graesser [DG12] 185
9.2 The resulting positive and significant affect transitions as compared to the D'Mello and Graesser [DG12] theoretical model. 191
9.3 The probability of a student persisting in each affective state over time. 195
9.4 The plotted exponential decay of each affective state as estimated by the sensor-free affect detectors. 197
9.5 The plotted exponential decay of each affective state as reported in Table 1 of D'Mello and Graesser [DG11] 198

10.1 The flowchart of possible student actions resulting in either quitting (refusal or stopout) or mastery of the assignment. 207

10.2 The frequency of student stopout by learning opportunity. Stopout on the first opportunity appears to be disproportionately larger than subsequent opportunities. 211
10.3 The exponential curve fit to stopout on the first ten learning opportunities. The line is a poor fit seemingly due to stopout on the first item. 213
10.4 The exponential curve fit to stopout on opportunities 2 through 10, extended to show predicted stopout on the first problem. 214
10.5 The resulting clusters of student prior correctness and last action pertaining to student stopout. 218
10.6 The proportional distribution of samples within each cluster over the first three learning opportunities. 220
10.7 The reported confidence of students within each cluster with associated 95% confidence intervals. 222
10.8 The reported confidence of students who stopout on the first learning opportunity as compared with students who stopout after the first learning opportunity with associated 95% confidence intervals. 223

11.1 The resulting structural equation model applied to the third dataset. 251
11.2 The distributions of each factor extracted from the detector models. . 252
11.3 The transformed distributions of each factor extracted from the detector models. 253

List of Tables

2.1 The bin student distribution and prediction values calculated for Fold 1 of Skill 47 of our dataset. 15

2.2 The overall percent correctness on the first response of all subsequent skills for each of the five bins. 16

2.3 Results of our trials over all skills 18

3.1 Comparing locations of dropout within the GRU model. 36

3.2 Three recurrent model variants, trained on both the resampled and non-resampled datasets, are compared to the previous highest reported results on the ASSISTments dataset. 37

3.3 LSTM model performance for each individual affect label. 39

4.1 Description of the generated action-level features. 50

4.2 The notable descriptives of the dataset. 53

4.3 Predicting Wheel Spinning in current assignment 59

4.4 Predicting Stopout in current assignment 59

4.5 Predicting Wheel Spinning in next assignment 61

4.6 Predicting Stopout in next assignment 61

5.1 The number of instances and distribution of labels across each outcome. 83

5.2 Comparison of feature sets across each of the detector models. 90

5.3 Comparison of feature sets across each of the detector models using co-training. All detectors in this analysis uses an LSTM model. *The machine learned model of each detector utilized co-training across actions and therefore mirrors the respective rows in Table 5.2. 93

7.1 Categories of detectors to be utilized by DRIVER-SEAT. 128
7.2 Participant timeline. 139
7.3 Potential actions to be utilized by DRIVER-SEAT 142
7.4 Conditions within the randomized controlled trial evaluating feedback development method, time allowed, and students' perceptions of message origin. 146

9.1 The transitions between affective states. D'Mello's L values are shown. Transitions that are statistically significantly more likely than chance, after Benjamini and Hochberg's post-hoc correction, are denoted *. . 202

11.1 The list of observed detectors of student knowledge, behavior, and affect. 236
11.2 The description of outcome measures 238
11.3 Counts of students, assignments, and classes across the three datasets. 240
11.4 The EFA factor loadings observing the student learning assessment measures. 246
11.5 The EFA factor loadings observing the detector models. 247
11.6 The CFA measures of fit for the EFA observing the detector models. . 249
11.7 The CFA measures of fit for the EFA observing the student assessment measures. 250
11.8 The beta coefficients, statistical reliability, and variance explained for each multi-level model. 255

11.9 The model results observing the distal outcome of TerraNova score. . 257

Chapter 1

Introduction

Failure is a difficult, yet inevitable and necessary step in the process of learning. Perseverance or persistence in the presence of failure is often considered the key to eventual success, having been studied through a myriad of research pertaining to grit [DPMK07], academic tenacity [DWC14], perseverance [PS+04][FRA+12], resilience [MW09], productive struggle [War15] and productive failure [Kap08]. In each of these models and theories, persistence alone is not enough to quantify a student's performance or effectively characterize their behavior. Student success is not simply a measure of whether that student made an attempt or quit, but also incorporates a measure of how much meaningful processing occurred during the learning activity [CGSG04]. This latent construct of learning, commonly described as cognitive engagement [WGM06] is likely a distinguishing factor that separates productive from unproductive persistence, or perseverance as these terms will be used interchangeably throughout this work, exhibited by students on a particular learning task.

But what makes persistence productive? It has become clear in studying learning that not all persistence is beneficial to a student and additional practice, particu-

larly when there is a gap in knowledge, may not help the student to progress toward understanding or mastery of the material. This case has been previously studied through a behavior known as 'wheel spinning' and describes when a student applies effort but exhibits little-to-no progress toward mastery of the topic [BG13]. However, it is necessary for students to exhibit some persistence when faced with struggle because giving up too soon may deprive the student of practice opportunities to either remedy the misunderstanding or gap in knowledge or, at the very least, provide assessment to identify the potential causes of the difficulty. This aspect of persistence has been studied through cases of student attrition as either 'dropout' [CRK15][XCSM16][YSAR13][RCY+14] describing a course or school-level attrition or 'stopout' [BVIH19] describing a lower, assignment-level attrition.

Wheel spinning and student attrition, henceforth referenced as student stopout to reflect the granularity of measurement observed in the current work, describe two aspects of unproductive persistence, but do not necessarily inversely describe productive persistence comprehensively. Surely, these two previously-studied behaviors are measures of unproductive persistence, but that does not mean that these are the only measures describing a lack of productivity. By definition, productive persistence describes cases where a student benefits from additional practice in regard to future learning, and as such, likely requires some level of cognitive engagement from the student as previously described. In forming the definition of productive perseverance, it becomes clear that there are numerous learning constructs involved and they likely interact in complex ways. Furthermore, the behaviors themselves can be observed at varying levels of granularity as engagement and persistence can be observed within a single learning task, across learning subtasks, or even at more longitudinal granularities across learning tasks.

To understand productive persistence, researchers must understand, identify, and

quantify student engagement and the surrounding behaviors. While the number of learning constructs that could describe student engagement is vast, this work focuses on three identified categories of measures: knowledge, behavior, and affect. While each of these describe inherently latent constructs of learning (i.e. one cannot directly observe the knowledge of a student), these constructs can be operationalized through various measures that are internally and often externally validated.

The adoption of computer-based learning systems in real classrooms has provided invaluable opportunities to study the learning process not only with fine levels of granularity, but also longitudinally and at large scale across multitudinous classrooms in various geographic settings and urbanicities. The depth and breadth of data recorded as students interact with these systems has allowed researchers to develop and evaluate learning theory in a manner that can help drive intervention and improve instruction.

Detectors of various student knowledge, behavior, and affect have been developed by combining existing learning theory with data collected from students interacting with computer-based learning environments. While some of these detectors are commonly associated with assessment measures of student knowledge (e.g. current or future correctness), others have focused on more behavioral (e.g. taking advantage of, or gaming the system), and even affective (e.g. boredom) attributes. While some research has utilized physiological sensors external to these learning systems (c.f. [ACB+09]), many instead rely only on the student interaction data and find patterns of activity that are found to correlate with externally validated measures of these various learning constructs; these later detectors, referred to as "sensor-free" detectors [PRB+16][BBH17], have greater potential for application at scale and across learning environments (both in terms of the learning system but also the physical environment in which students work) as they do not require the installation

of additional sensors which may be both costly and also intrusive.

By distinguishing between productive and unproductive perseverance, it is the hope that this dissertation can facilitate much broader impacts in the study of student knowledge, behavior and affect and lead to actionable understanding of these constructs. The concepts that inspire this dissertation follow previous education research and learning theory to develop measures and detectors of student learning. This research promotes a better understanding of the relationship between such measures and detectors, and using these relationships to develop interventions aimed to support positive learning practices.

This dissertation is divided into three parts that address the development of sensor-free detectors of student behavior and affect, the development of tools to support action and intervention, and finally the deeper exploration of the detector models in the context of identifying relationships in the measured underlying constructs of learning. Throughout these parts, the chapters include a collection of publications, manuscripts in submission, two submitted federal grants (one of which that has been funded as NSF #1822830), and in-preparation works. Where applicable, the chapter title is accompanied by the associated abstract and citation with the author listing in acknowledgement of the invaluable contributions of all co-authors.

Part I

Developing, Improving, and Applying Student Models to Study Learning

Chapter 2

The Prediction of Student First Response Using Prerequisite Skills

Botelho, A., Wan, H., & Heffernan, N. (2015, March). The prediction of student first response using prerequisite skills. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, 39-45. ACM.

Abstract

A large amount of research in the field of educational data analytics has focused primarily on student next problem correctness. Although the prediction of such information is useful in assessing current student performance, it is better for teachers and instructors to place attention on student knowledge over a longer period of time. Several researchers have articulated that it is important to predict aspects that are more meaningful, inspiring our work here to utilize the large amounts of student data available to derive more substantial predictions over student knowledge. Our goal in this paper is to utilize prerequisite information to better predict student knowledge quantitatively as a subsequent skill is begun. Learning systems like ASSISTments and Khan Academy already record such prerequisite information, and can therefore be

used to construct a method of prediction as described in this paper. Using these inter-skill relationships, our method estimates students initial knowledge based on performance on each prerequisite skill. We compare our method with the standard Knowledge Tracing (KT) model and majority class in terms of the predictive accuracy of students first responses on subsequent skills. Our results support our method as a viable means of representing student prerequisite knowledge in a subsequent skill, leading to results that outperform the majority class and that are comparably superior to KT by providing more definitive student knowledge estimates without sacrificing predictive accuracy.

2.1 Introduction

A large amount of research in the field of educational data analytics has focused primarily on student next problem correctness. Events such as the Knowledge Discover and Data Mining Competition held in 2010 (*www.kdd.org*), more commonly referred to as the KDD Cup, directs the attention of the field to the prediction of next problem correctness; while perhaps useful in performance evaluation, the ability to predict next problem correctness has certain limitations in regards to utility especially when assessing student knowledge over larger periods of time. Others in the field have begun raising other meaningful questions[BmCMC08] [PH10b][WH13], realizing the importance of predicting or observing aspects that are much more substantial. Intelligent tutoring systems (ITS) provide a wealth of student data from which more meaningful predictions and observations can be derived. Our work here aims to utilize this data to provide more significant information pertaining to student knowledge to teachers and instructors.

For our research, we emphasize constructing a more precise prediction on students' initial knowledge approaching a new skill. In the general case, students move

gradually from an initial state of knowledge toward mastery, and student models should capture this change. Thus, a more accurate estimation of this initial knowledge could lead to a better understanding of a student's knowledge state at any observable time, and consequently, we could use the model's results to develop more precise predictions of future performance.

In this paper, we utilize prerequisite information to predict student initial knowledge on subsequent skills. If a skill 'A', is a prerequisite of skill 'B,' students should have mastered 'A' before proceeding to 'B.' The prerequisite relationships used in this work are defined by domain experts. Due to human effect, some skill relationships might be overestimated, or they may not exist in other applications. As such, we are seeking to answer the following two questions in this paper:

1. Does prerequisite information really help to improve the estimation of initial knowledge on subsequent skills?

2. Are all prerequisite relationships reliable?

We address these questions through three experiments to first observe trends of distribution across our proposed binning method, and then to compare the predictive accuracy of that method to that of KT and majority class across all skills and at an individual skill level.

The next section will introduce a background of our comparative model, KT, after which we will described the dataset used in our trials. The following section will discuss our proposed binning methodology before illustrating the results of our experiments, and, finally, we state our contributions, conclusions and intended future work in this field of research.

2.2 Background

The knowledge tracing (KT) model [CA95] developed by Corbett and Anderson has long been successful in the field of student assessment. Its implementation and use in tutoring systems and use in performance analysis systems exemplifies its practical applications, scalability, and appropriation across many fields of study. The KT model is widely used in these tutoring systems and the field of educational data analytics due to its accuracy in predicting student correctness by utilizing only a small amount of data.

The KT model gains its accuracy through the training of four parameters representing students' prior knowledge, learning rate, probability of guessing, or answering correctly while not knowing a skill, and chance of slipping, or answering incorrectly while in a supposed "learned" state. Knowledge tracing relies heavily on the successful training of these parameters to properly model a student beginning a new skill, and then to build upon that model at a student level given a sequence of responses. For this reason, each student beginning a particular skill receives the same base model. Therefore, each student within a skill will be given the same prediction for the first response. The model could be greatly improved if another prediction procedure, such as the method described in this work, could use a more intelligent approach to predict first response.

In the standard KT model, initial knowledge is represented by a parameter $P(L_0)$, the probability of mastering the skill [CA95]. As such, KT is often used to estimate each student's initial knowledge [PH10b]. In the standard KT model, the parameter $P(L_0)$ is trained on all students' records in the a training set, and assumes that all students have the same initial state of knowledge. However, this assumption is too strong to use the model to predict each individual student's first

response. To overcome this drawback, Pardos and Heffernan use three heuristic functions to model individualization in KT [PH10a], and find that the method, setting initial individualized knowledge based on individual students' performance over all skills, yields superior results. This approach, however, overestimates the relationships between skills. If learning a skill does not promote, or even hinder [CM], learning another skill, then it is not appropriate to use knowledge in one skill to estimate another.

Baker et al. uses another method [dBCG$^+$10] that compares a student's overall performance and all other students' performance on a skill to build an individualized model. Like the standard KT model, this method suffers two major problems: falling into local maxima and the existence of multiple global maxima[BmC07]. Thus, we cannot know if the value of P(L_0) obtained by the model represents true student initial knowledge.

Knowledge Tracing's many strengths have made it a kind of comparative model in many works and is used again here as such. Knowledge Tracing builds upon the performance history of each student to calculate a probability that the student will answer the next problem correct. For this reason, it often fails to accurately predict students' first responses as there is less information for KT to accurately calculate a prediction. The method of prediction proposed in this work focuses entirely on first responses of students undertaking a new skill by observing student performance in prerequisite skills. Using knowledge tracing as a comparative model, our method of prediction aims to outperform KT in terms of accuracy in regard to students' first responses while providing a more definitive measure of initial knowledge.

2.3 Dataset

The dataset used in our work is comprised of real-world algebra and geometry-based student data from the 2009-2010 academic year taken from the ASSISTments tutoring system. This system is administered by teachers to students through assigned problem sets that track student performance in addition to many other features to be used for better assessing each student's knowledge and understanding of each topic, or skill. It is intended that each student completes problems pertaining to the assigned skill until a status of mastery is reached, which by default is defined as three consecutive correct answers. Each problem, or opportunity as it will be referred to in this work, is recorded by the system and is used to evaluate that student's overall performance.

Within ASSISTments, skills are arranged in an intended prerequisite-to-subsequent skill structure defined by domain experts as a recommended sequence of topics for instructors. It is the teacher's choice which skills and problems to assign as well as the order in which to assign them. As will be discussed later, the relationships between skills in this predefined prerequisite structure is worth further inspection, but are trusted for our initial experiments.

It was found that of the 230 skills listed as subsequent skills in our ASSISTments dataset, 28 contained usable prerequisite data; we define student data as usable if the sequence in which students complete skills matches the prerequisite structure defined within the system. The usable dataset consisted of 983 unique students across all skills, providing 3466 rows of response data. We acknowledge that our results may provide more reliable conclusions with a larger dataset, but our work here is intended to be used as initial work from which further research may expand upon and is therefore viewed as sufficient for this paper.

From the student performance, we also calculate each student's individual speed of mastery, defined as the number of opportunities, or individual problems, completed in order to gain mastery status as described above. We use this mastery speed as a measure of student knowledge and aptitude across skills and is used to calculate predicted responses as described in the next section. For this work, only problem correctness, expressed as binary values in the system, is used to calculate mastery speed and overall student performance, neglecting other features such as time between problems and skills and also partial credit evaluations. Other methods of determining mastery, discussed briefly in a later section, may lead to improved accuracy in our method, but are not the focus of this work; we use the simple "three right in a row" method of determining mastery for all of our experiments.

2.3.1 Methodology

The method described in this work attempts to better predict student first problem correctness on a subsequent skill by categorizing, or "binning", students with similar mastery speed in a prerequisite skill; for purpose of clarification, the terms bin and category will be used interchangeably throughout this paper. Such a method has shown success in the past [XLB13] when making other predictions such as next problem correctness using different features from a similar dataset. This method, labeled as "Prerequisite Binning" (PB), involves categorizing students based on a set of features, such as mastery speed, and inferring a relationship between them. For example, we binned students with similar ranges of mastery speed in order to create a prediction for any student that also could be placed in the same bin. If successfully identified, certain trends may appear within the bins, which are addressed in a later section.

The method of binning, as mentioned, groups students based on prerequisite

Student	Prerequisite	Mastery Speed	Skill	First Response Correctness
Tom	Adding	4	Division	Correct
Tom	Mult.	8	Division	
Bill	Adding	3	Division	Incorrect
Bill	Mult.	6	Division	
Joe	Adding	3	Division	Correct
Joe	Mult.n	3	Division	
Sue	Adding	5 LPC	Division	Incorrect
Sue	Mult.	DNF LPC	Division	

⇓

Attempts	Prediction	Number of Students
3-4 incl.	1.0	1
4-8 excl.	0.5	2
8+	0.0	0
DNF High % Cor.	0.0	0
DNF Low % Cor.	0.0	1

Figure 2.1: The hypothetical students and data shown, fabricated to show our methodology, exemplifies the table creation process. Using a training set, a probability table is created for each skill by categorizing students with similar performance history

mastery speed. For this, we used a 5 fold cross-validation on our dataset, using 80% as a training set to predict performance on the remaining 20%. The training set was used to construct the bins, which splits students based, again, on the number of opportunities needed to master each prerequisite skill. An average mastery speed across all prerequisite skills was calculated, placing students into one of five bins. The first bin contains those who averaged three to four opportunities inclusively ($3 \leq x \leq 4$) to master all prerequisite skills; as three opportunities is the lowest possible mastery speed and four opportunities indicates an incorrect response on only the first problem, this bin presumably represents the highest knowledge students. The second bin, following the first in terms of mastery range, contains students who require, on average, between four and eight opportunities exclusively ($4 < x < 8$).

The third bin encompasses students with an average mastery speed of eight or more ($8 \leq x$) across all prerequisite skills. Following this categorizing strategy, a fourth bin would contain those students that did not reach mastery status on prerequisite skills before proceeding to the subsequent skill. However, our dataset shows that a large percentage of students fall into this category, many of which respond to only a small number of problems; the reason for neglecting to finish a particular skill could be explained by boredom, simple negligence, poor time management, or a lack of knowledge. For these reasons, the "did not finish" (DNF) category, describing students that did not master all prerequisite skills, is represented by two bins. Our fourth bin contains students that did not master at least one of the prerequisite skills with a high percent correctness (HPC) across those skills (greater than or equal to 66.67% correctness), while the fifth and final bin contains such students with a low percent correctness (LPC) across all prerequisite skills (less than 66.67% correctness). The fourth and fifth bins handle the case where a student began a prerequisite skill, but did not reach mastery status; this means that at least one problem was attempted, but the student either completed less than three or failed to answer correctly on three consecutive opportunities. Bin four is therefore meant to represent students that failed to complete the prerequisite skills for reasons other than lack of knowledge, while the fifth contains students genuinely struggling and are perhaps experiencing "wheel spinning"[BR14].

With students from the training set categorized based on performance in prerequisite skills, a prediction value was calculated for each bin by finding the percentage of students in each category to respond correctly on the first opportunity of the subsequent skill. The reasoning for this method of binning, again, stems from the theory that particular trends exist for students in each bin and will extend to other students that also fall into that category. Therefore, it was expected that the pre-

diction value of each bin constructed by the training set would apply to similar students in the test set.

Bin	Num. of Students	Num. of Students with First Response Correctness	Bin Prediction
1	29	24	0.828
2	53	26	0.491
3	3	0	0.000
4	2	1	0.500
5	3	0	0.000

Table 2.1: The bin student distribution and prediction values calculated for Fold 1 of Skill 47 of our dataset.

Figure 2.1 exemplifies the bin creation process using a set of hypothetical students (the names and values do not reflect any real person/dataset and are purely exemplary). In that example, prerequisite information from four students is used to construct the five bins. As Tom averaged a mastery speed of 6 opportunities across all his prerequisite skills, he is placed into the second bin with Bill, who averaged a mastery speed of 4.5 opportunities. Since Tom answered correctly on the first problem of the subsequent skill and Bill did not, the prediction for the second bin becomes 0.5, as half of the students in that bin answered the first problem of the subsequent skill correctly. Joe mastered each prerequisite skill with the minimum three attempts and is therefore placed into the first bin. That bin is given a probabilistic prediction of 1.0 due to the fact that all students in that bin answered correctly on the first question of the subsequent skill. Sue is placed into the fifth bin, as she did not master one of the prerequisite skills, and had a low percent correctness (less than 66.67%) across both prerequisites. She did not answer the first problem correct on the subsequent skill, leading to a prediction of 0.0, as no student in that bin answered correctly on the first question of the subsequent skill.

The values depicted in Table 2.1 were generated from our actual dataset. This

table illustrates the prediction calculation methodology using skill 47 of our dataset corresponding in the ASSISTments tutoring system to the "Conversion of Fraction Decimals Percents." As described in the earlier example, students in a training set are placed into each bin based on estimated student knowledge. Using this categorization, a prediction is calculated by observing the number of students in each bin to answer the first problem of the subsequent skill correctly.

2.4 Results

The results of our work are exemplified through several metrics. Before comparing the predictive accuracy of our binning method to any other model, we must verify that each bin represents the intended level of knowledge within our dataset. Our method is able to illustrate this representation by observing the percentage of students within each bin to answer correctly on the first problem of a subsequent skill.

Bin	**Number of Students**	**Percent Correct on First Response**
1	806	61.79%
2	1170	60.00%
3	172	54.65%
4	732	52.59%
5	586	50.51%

Table 2.2: The overall percent correctness on the first response of all subsequent skills for each of the five bins.

Table 2.2 shows the distribution of knowledge within each bin across all skills in the observed dataset. The values show a distribution of higher knowledge students in the lower bins and lower knowledge students in the higher bins. This result supports the claim that our method is properly representing the intended level of knowledge.

The distribution of the number of students in each bin, particularly the fourth and fifth bin, indicates that our dataset contains a large percentage of students that did not complete prerequisites before attempting a subsequent skill. This was the reasoning behind splitting this "DNF" bin into a high knowledge and low knowledge bin based on percent correct in the prerequisite skills. Further splitting these bins may lead to better predictive accuracy of our method, but is sufficient for our work in its current state and avoids over-complicating what is meant to be a simple categorization method.

2.4.1 Comparison of Overall Performance

The results of our method, entitled "Prerequisite Binning" in Table 2.3, was compared to knowledge tracing as well as a majority class (MC) prediction to act as a control in our experiment. We chose knowledge tracing as it is widely used and studied in the field of educational data analytics and attempts to learn student initial knowledge for use in its calculation. Through this experiment we are first observing the effectiveness of our model by comparing it to the majority class, a prediction made for all students using the average correctness of the dataset, and then observing the differences in error between our method and KT; results illustrating a comparable error between the two methods supports the use of our binning method over KT, as it provides more definitive estimates of student knowledge without sacrificing predictive accuracy. Knowledge tracing was run using Kevin Murphy's Bayes Net Toolbox for MATLAB [Mur] with initial parameters of 0.30, 0.14, 0.20, and 0.08 for prior, learn, guess, and slip respectively. For our experiment we ran a five fold cross validation on our dataset, using 80% of the data from each skill as a training set to predict the remaining 20%. The results in Table 2.3 represent the averages of all folds for each method.

Each of the three prediction methods are compared using RMSE and AUC two common measurements of error. A low RMSE indicates a more accurate prediction method while a larger AUC indicates higher accuracy. As observed in Table 2.3, the prerequisite binning method outperforms the majority class in both metrics indicating that it is a successful prediction method. When compared to knowledge tracing, however, the results show nearly the same RMSE value, but a superior AUC value.

While the binning method may not outperform knowledge tracing in all metrics, the predictive accuracy is comparable. The purpose of this work, again, is not to provide a method that outperforms KT, but rather to construct a modeling method that can provide teachers with more meaningful information regarding student knowledge. Unlike KT, where the learned parameters such as prior/initial knowledge are unusable metrics in describing true student knowledge due to the identifiability problem [BmC07], our binning method provides an initial knowledge estimate based on previously observed performance; this initial knowledge estimate, represented as the probabilistic prediction calculated for each bin, is shown to be just as reliable as KT in predictive accuracy, while also providing a more definitive metric to describe a bin-wide initial knowledge that avoids problems of identifiability.

	RMSE	**AUC**
Majority Class	.496	.570
KT	.472	.626
Prerequisite Binning	.473	.651

Table 2.3: Results of our trials over all skills

Based on the results of our trial, we can conclude that prerequisite information can be used to predict student performance on subsequent skills in regards to first re-

sponse. This supports our argument that knowledge and learning can be observed between prerequisite and subsequent skills.

2.4.2 Comparison Over Individual Skills

We also compare our method with KT on each individual skill. Figure 2.2 shows the difference of RMSE for these two models, that is: RMSE(KT) − RMSE(Bin); each positive difference value, therefore, indicates that our binning method outperforms KT in that skill, while negative difference values indicate KT outperforms binning in that particular skill. Each bar in the figure has an accompanying p-value above. This p-value is computed by applying a statistical T-test on the five-fold cross validation results. From this figure, we observe that our method outperforms KT in 14 of the 28 observed skills. Looking at the T-test results, there is a significant difference (p-value ≤ 0.05) between the two models on only 3 skills. This statistic further supports the comparability of the two models in terms of accuracy.

A similar histogram illustrating the difference of RMSE for the majority class and our binning method, RMSE(MC) − RMSE(Bin), can be seen in Figure 2.3. The majority class represents a prediction for each student that is equal to the percent correctness of the training set of students. Again, as we used a five fold cross validation, 80% of the data from each skill is used as a training set to predict the remaining 20%. Comparing our binning method to the majority class should provide results that take into account the difficulty of each skill, defined by the average correctness calculated in majority class predictions.

This result attempts to answer the second question in introduction pertaining to the reliability of the prerequisite skill relationships. In accordance with our initial thoughts, the stronger the relationship between a prerequisite and subsequent skill,

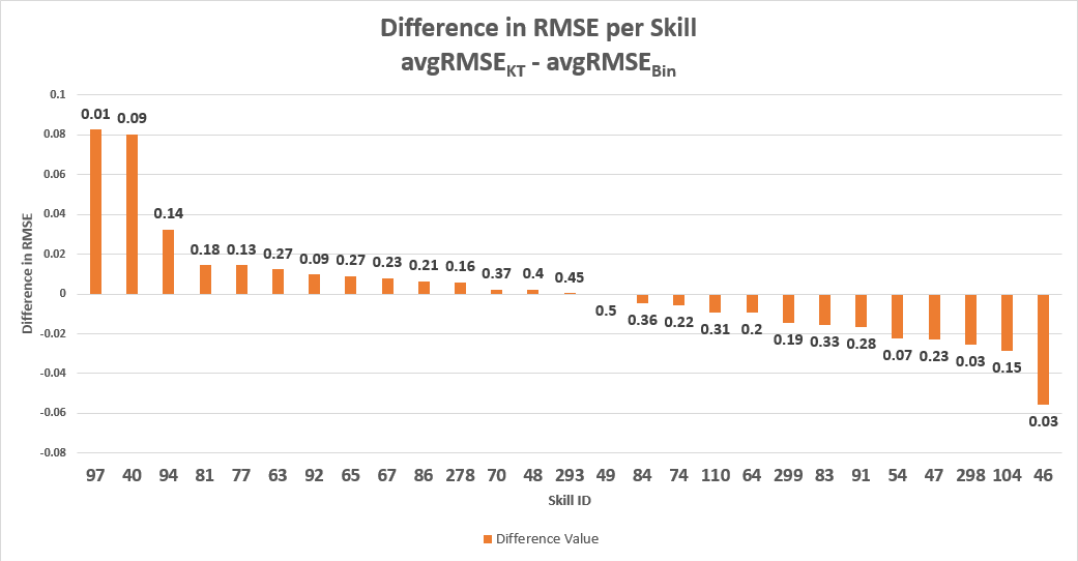

Figure 2.2: The difference of RMSE per skill when comparing our method of binning to standard knowledge tracing, ordered from highest to lowest difference. The number above each skill indicates the p-value of the difference.

the better we can predict the performance of the subsequent skill from the knowledge of the prerequisite skill. Using Figure 2.3, we can observe significant differences (p-value ≤ 0.05) in terms of RMSE on a total of 5 individual skills. Therefore, at least on skills 97 and 49, the skills with better statistically significant results, we have strong confidence that the prerequisite relationships are reliable. For those skills with significantly lower results, skills 54, 298, and 46, the causal relation of the prerequisite skills may not be strong as expected by domain experts. All other skills, however, do not illustrate results significant enough to make a claim. These particular inconclusive results may be explained by inspecting our dataset. As indicated in our first observations pertaining to the distribution of students in each bin, a large percentage of students are categorized into bins four and five. Many of those students, as indicated by our dataset, attempt less than three problems, preventing mastery and also making it more difficult to properly estimate knowledge.

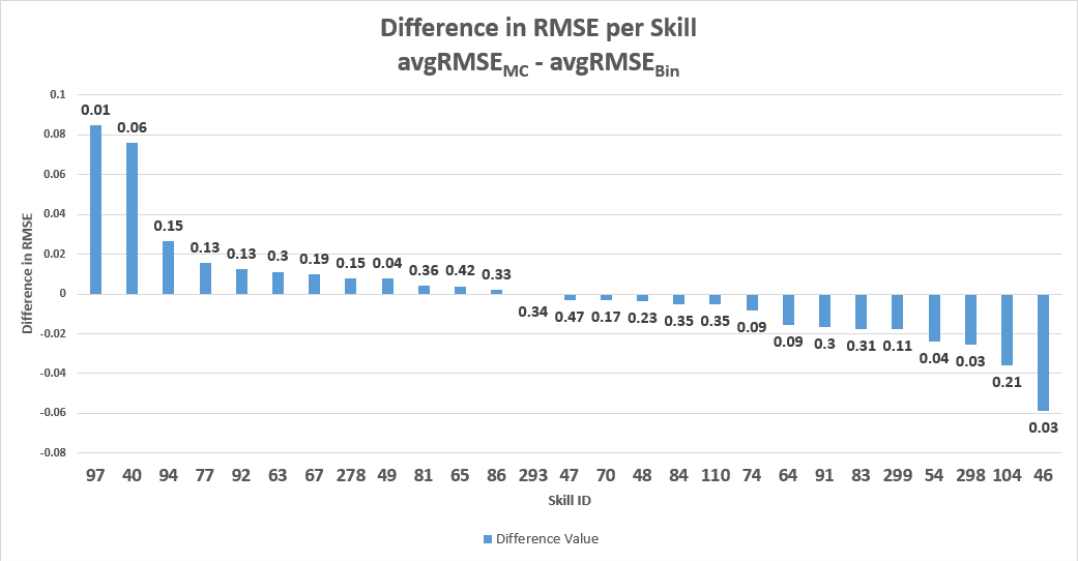

Figure 2.3: The difference of RMSE per skill when comparing our method of binning to majority class predictions, ordered from highest to lowest difference. The number above each skill indicates the p-value of the difference.

There may be two reasons for this occurrence. First, the prerequisite skills may too hard for the students to master. This may result from the teacher's decision not to assign particular prerequisite skills, or the skill relationship graph is incomplete. A second possibility may allude to a case where a teacher does not assign enough questions for students to master the prerequisite skills. As a teacher has control over the administering of skill problems, a number of such scenarios could lead to such results. In summary, these findings potentially indicate a need to further inspect the causal relationships defined by domain experts as they appear in the observed systems.

2.5 Contribution

Our goal in this paper was to utilize the prerequisite information that many systems record to infer aspects of the students in our data. The current predominantly

used knowledge tracing model employed in many learning systems assumes that all the skills are independent of each other. In this work, however, prerequisite information is used to better understand the relationship between the prerequisite and subsequent skills. The added consideration of this relationship in a model can be used to make better statements and inferences about not only the students, but also in the manner that such skills are presented to students.

We have shown here, to our knowledge, the first model that attempts to use the relationships between prerequisite skills to predict subsequent knowledge. This is on its way to make a larger contribution to better personalizing and individualizing student models by acknowledging and utilizing more of the data. We will make note that there are many other researchers that have used aggregate information, but have not paid attention to the prerequisite structure. Many psychometricians have found, for instance, that if students who do well on a topic A tend to do well on a topic B, that information can be used to better predict performance on topic B. In this context, however, we prefer to view such information differently. Our ultimate goal is to be able to make statements to teachers regarding information that is more causally related, and we do not want to influence predictions of future performance for unrelated tasks where there is little knowledge overlap. By imposing this constraint upon us, it will reduce our ability to make predictions, but will increase the significance of our statements to teachers.

The goal of this paper extends beyond the intent to develop a more accurate prediction methodology. We wish to look at the causal effects from which our results derive. It is more of a question of why using this data from prerequisite skills produces the accurate predictions across some skills and not in others. Our findings support the intuitive claim that certain skills are related, while others are not. Our trials provide a means of visualizing aspects of such skills to show that, as in

Figure 2.3, prerequisite information does not have the same effect for all subsequent skills. Observing little difference in some skills between a method utilizing prerequisite information and a method, such as KT or majority class, that does not use such information may point to several issues in either our dataset or the prerequisite graph of the system. It is an interesting observation that some skills, while listed as a prerequisite, may not have as strong a relationship to a subsequent skill, which is vitally important information to teachers who need to consider a sequence to introduce new skills.

2.6 Conclusion and Future Works

The results and observations presented in this paper open new research opportunities in student assessment. Through our results we have observed several factors that help to better model student knowledge and aptitude across skills. The trials of this paper certainly raise some curiosities as to the extent subsequent skills are affected by prerequisite performance. In this paper, we focus exclusively on first responses of subsequent skills and, as the results were successful, we can now look beyond the first response to observe trends in prerequisite influence over an entire subsequent skill response sequence.

With these findings, our method can be adapted and/or appropriated to benefit other models like KT. Implementing our method into a modified KT model could lead to more accurate representations of student initial knowledge. As the method we propose here requires little in terms of processing time while providing more definitive student knowledge estimates than other models like KT, we aim to, through similar methods, represent other aspects of student learning such as aptitude and knowledge retention.

The accuracy of this method of binning is largely impacted by the reliability of the method of determining mastery. In our experiments, as it is in ASSISTments, mastery is defined simply as a student answering correctly on three consecutive opportunities; this method, while simple, may not be the best means of representing such a status universally for all students. Further work in exploring more precise methods of determining mastery speed may prove to benefit our method; such a method may include the individualization of mastery speed requirements for each bin, as it is likely that not all student levels of knowledge can be confidently labeled as having mastered a skill with the same number of sequential correct responses.

In this work, we concern ourselves with and direct our attention to the concept of student growth and knowledge over time. We believe that such information identifies aspects of the student more definitively than next problem correctness. In the future, we hope to continue similar work, looking into the influences that prerequisite skills exhibit in the other student models, like the wheel spinning model [BR14]. We would also like to make further observations and inferences on prerequisite skills, such as their impact on the student learning process itself, or the retention performance [XBL13] of this prior.

Chapter 3

Improving Sensor-Free Affect Detection Using Deep Learning

Botelho, A. F., Baker, R. S., & Heffernan, N. T. (2017, June). Improving Sensor-Free Affect Detection Using Deep Learning. In *International Conference on Artificial Intelligence in Education*, 40-51. Springer, Cham.

Abstract

Affect detection has become a prominent area in student modeling in the last decade and considerable progress has been made in developing effective models. Many of the most successful models have leveraged physical and physiological sensors to accomplish this. While successful, such systems are difficult to deploy at scale due to economic and political constraints, limiting the utility of their application. Examples of "sensor-free" affect detectors that assess students based solely using data on the interaction between students and computer-based learning platforms exist, but these detectors generally have not reached high enough levels of quality to justify their use in real-time interventions. However, the classification algorithms used in these previous sensor-free detectors have not taken full advantage of the newest methods

emerging in the field. The use of deep learning algorithms, such as recurrent neural networks (RNNs), have been applied to a range of other domains including pattern recognition and natural language processing with success, but have only recently been attempted in educational contexts. In this work, we construct new "deep" sensor-free affect detectors and report significant improvements over previously reported models.

3.1 Introduction

While intelligent tutors have a long history of development and use, the most widely-used systems remain less sophisticated than initial visions for how they would operate. The systems now used at scale are often cost-effective and have been shown in large-scale randomized controlled trials to lead to better learning outcomes (e.g. [PGMK14],[RFMM16]), but do not reach the full level of interactivity of which human tutors are capable. For example, one positive aspect of human tutors is the ability to observe student affective state and adjust teaching strategies if students are exhibiting disengaged behavior [LMDP08]. Student emotion and affective state have been found to correlate with academic performance [CGSG04][PBSP+14] and can even be used to predict which students will attend college [PBBH13].

With increasing evidence supporting the benefits of utilizing student affective state to drive tutoring strategies [DLS+10], it is important to develop accurate means of detecting these states from students working in these systems. While strides have been made to build accurate detectors, many successful approaches include the use of physical and physiological sensors [ACB+09][DLS+10][PRB+16]. However, it can be impractical to deploy such sensors to classrooms at scale, both for political and financial reasons. Detecting affect solely from the interaction between the student and learning system, sometimes referred to as sensor-free affect detection, may be

more feasible to deploy at scale. However, while these models' predictions have been usable in aggregate for scientific discovery, the goodness of these approaches has often been insufficient for use in real-world intervention.

Sensor-free affect detectors have existed for several years and have been used to assess student affective states using low-level student data as students interact with a mouse and keyboard [SMSB14] , but also using features extracted from a range of learning platforms including Cognitive Tutor [dBGW$^+$12], AutoTutor [DCW$^+$08], Crystal Island [SML11], and ASSISTments [OBG$^+$14][WHH15]. While these detectors have been better than chance, their goodness has fallen short of detectors of disengaged behavior, for example (cf. [PBSP$^+$14]). Increasing the accuracy of sensor-free affect detectors would lead to higher confidence in their use to drive intervention.

In this paper, we attempt to enhance sensor-free affect detection through the use of "deep learning," or specifically, recurrent neural networks (RNNs) [WZ89]. Previous affect detectors have utilized a range of algorithms to detect student affective state; we study whether deep learning can produce better predictive accuracy than those prior algorithms. We study this possibility within a previously published data set to facilitate comparison with and understanding of the benefit derived from using this algorithm. Recurrent neural networks are a type of deep learning neural network that incorporates at least one hidden layer, but also provides an internal hidden node structure that captures recurrent information in time series data.

RNNs are most appropriately applied to time series data, where the output of the current time step is believed to be influenced or impacted by previous time steps. In this way, it is believed that affect detection could benefit from a model that observes the temporal structure of input data. Several internal node structures have been proposed, yielding variants of traditional RNNs such as Long-Short Term

Memory networks (LSTMs) [HS97] and more recently Gated Recurrent Unit networks (GRUs) [CVMBB14]. Applications of these deep learning algorithms have been used in other domains for pattern recognition [CGCB14] and improving natural language processing [SLMN11]. Performance in these domains certainly suggest large benefits in using deep learning on temporal or time series information.

Deep learning prediction models have not yet been used extensively in educational domains, but have been studied as a potential method to improve the decisions of virtual agents in game-based learning environments [MWP+16] and also to improve the prediction of student correctness on the next problem [PBH+15]. However, the results of the "Deep Knowledge Tracing" (DKT) model presented in [PBH+15] are as yet uncertain; initial reports suggested profoundly better performance than previous approaches, but later investigation by other researchers indicated that the same data points were being replicated and used to predict themselves, artificially inflating goodness [XZVIB16]. When this error was corrected, performance seemed to be equivalent to previous approaches [KLM16]. Nonetheless, recurrent neural networks may be highly effective for problems with the complexity and the quantity of data available to fully leverage their benefits.

As such, this work seeks to apply deep learning to utilize student information to better detect students' affective states without the use of sensors. We explore the application of recurrent neural networks for the task of detecting affective states using data collected in the context of the ASSISTments online learning platform.

3.2 Dataset

The dataset[1] used to evaluate our proposed deep learning approach to detecting affective state is drawn from the ASSISTments learning platform [HH14]. ASSISTments is a free web-based platform that is centered around providing immediate feedback to the many students who use it in the classroom and for homework daily. ASSISTments also provides on-demand hints and sequences of scaffolding support when students make errors. The system was used by over 40,000 students across nearly 1,400 teachers during the 2015-2016 school year, and has been found to be effective in a large-scale randomized controlled trial [RFMM16].

3.2.1 Data Collection and Feature Distillation

The ground truth labels used in this dataset come from in-class human observations conducted using the Baker-Rodrigo Ocumpaugh Monitoring Protocol (BROMP) [OBR15]. These quantitative field observations (QFOs) were made by trained human coders who observed students using the ASSISTments learning platform in a classroom environment. The coders observed students and labeled their affect as bored, frustrated, confused, engaged concentration, or other/impossible to code. They collected affect observations over 20-second intervals in a round-robin fashion, cycling through the entire class between observations of a specific student. Unlike approaches using video coding or retrospective emote-aloud (e.g. [CDWG08]), this approach inherently leads to missing labels between observations of the same student. These missing intervals for each student are known, as timestamps are recorded for each observation, and will be taken into account when formatting the data for input into the recurrent neural network; this process is described in more

[1] Our dataset is made available at http://tiny.cc/affectdata

detail in a later section.

A total of 7,663 field observations were obtained from 646 students in six schools in urban, suburban, and rural settings. In prior work [WHH15], a set of 51 action-level features was developed using an extensive feature engineering process; these features consist of within- and across-problem behaviors including response behavior, time working within the system, hint and scaffold usage within the system, and other such features attempting to capture various low-level student interactions with the system. As the observation intervals, or clips, often contain more than one student action within the learning system, the features were aggregated within each clip by taking the average, min, max, and sum of each feature. The end result was 204 features per clip.

In this paper we will compare our deep learning-based detectors of student affect to two earlier sensor-free models of student affect within ASSISTments (e.g. [OBG+14][WHH15]). In doing so, we will use the exact same training labels and features as in [WHH15], in order to focus our comparison solely on the use of deep learning.

3.3 Methodology

We input these labels and features into three deep learning models representing three common variants of recurrent networks including a traditional recurrent neural network (RNN), a Gated Recurrent Unit (GRU) neural network, and a Long-Short Term Memory network (LSTM). The GRU variant was chosen when exploring network structures and hyperparameters for training for both its faster training times in comparison to the LSTM variant and also for its increased ability to avoid problems such as vanishing gradients to which traditional RNNs are more susceptible.

The models explored in this work were built in python using the Theano [TARA+16] and Lasagne [DSR+15] libraries.

3.3.1 Network Structure

Our implementations each use the same three layer design, with an input layer feeding into a hidden recurrent layer of 200 nodes, progressing to an output layer of four nodes corresponding to each of four classes of affective state. The input layer accepts a student-feature vector of 204 generated covariates per time step normalized using the mean and standard deviation of the training set, and each network ultimately outputs 4 values representing the network's confidence that the input matches each of the four labels of engaged concentration, boredom, confusion, and frustration. A rectified nonlinear activation function is used on the output of the hidden layer, while a softmax activation function is used for the final model output.

Due to the large number of parameters present in deep learning networks, it is common to implement techniques to avoid overfitting. We adopt the common practice of incorporating dropout [SHK+14] into our model, which, in a general sense, sets some network weights to 0 with a given probability during each training step. This creates a changing network structure in terms of its interconnectivity during training to help prevent the model from relying on just a small number of input values. In our three layer model, dropout can be applied before and/or after the recurrent layer, and this is explored to determine which location of placement produces superior performance. We incorporate 30% dropout, such that each weight in the network, in the location dropout is applied, has a 30% chance of being dropped for a single training step; many implementations instead describe dropout in terms of a "keep" probability, but is described here as a "drop" probability to remain

consistent with the library used to build the models. As is standard practice, dropout is not used when applying the model to the test set.

3.3.2 Handling Time Series Data and Labels

The dataset used for the previous detectors in ASSISTments, and again in this work, consists of 20 second interval clips to which an affect label has been applied. The recurrent network takes as input a sequence of these clips to make use of the recurrent information within the sequence. The labeled clips, however, are not consecutive due to the design of the field observations, leading to gaps in student observations; during a gap in one student's sequence, the human coders present in the classroom were observing other students. It is possible to represent the non-consecutive clips as a full sequence, however, treating clips that are distant in time as consecutive may confuse the network and reduce performance. For this reason, we treat clips as consecutive only if they occur within 5 minutes of the previous labeled clip. Clips that occur beyond this threshold form a new sequence sample, resulting in a larger number of samples consisting of shorter sequences.

Another issue presented by the classification task is the non-uniformity of the distribution of the labels. The vast majority, approximately 80% of the clips, are labeled as engaged concentration, followed by 12% labeled as boredom, and only 4% each of confusion and frustration. While it is perhaps encouraging to know that students are mostly concentrating when working within ASSISTments, a model trained with labels in such non-uniformity may bias in favor of the more frequent labels. While it is often beneficial for the model to understand this distribution to some extent, it is better for the model to learn the trends in the data that correspond to each label rather than simply learn the overall distribution.

The original, non-recurrent affect detectors corrected for this issue by resampling

each of the labels [PBSP+14], but this cannot be directly reproduced here due to the time-series input into the recurrent network. In that previous work, the training data was sampled with replacement proportional to the distribution such that the resulting dataset is balanced across the distribution of labels and then evaluating on a non-resampled test set [EJJ04]. Rather than representing each sample as independent as in previous detectors, the recurrent network observes a sequence of observations within a single training sample. As such, we resample entire sequences including rarer affective states. Resampling in this way is likely to also resample the other labels as well, particularly when resampling the more scarce labels of frustration and confusion. While it is difficult to achieve perfect uniformity, sampling with replacement is performed using a threshold to balance the labels to a feasible degree. In this way, each sample of the training set is selected at least once, duplicating only those sequences containing at least 20% of one of the less common labels. From the resulting resampled data, we randomly downsample to the size of the original non-resampled training set for faster training times; training on the full resampled dataset did not produce substantial gains in model goodness over using the downsampled training set.

In an effort to further account for the non-uniformity of the distribution of labels, a final normalization is applied to the output of the network. The training data is used to determine the minimum and maximum prediction values for each label that is then used to scale the resulting predictions during model evaluation to span the entire 0 to 1 range (any prediction values in the test set outside of this range are truncated). This rescaling helps to deter the model from making overly conservative estimates of the less frequent labels. The output normalization is found to be necessary in this regard as estimates for the scarce labels rarely surpassed a 0.5 rounding threshold after the softmax activation of the output.

3.3.3 Model Training

All models are evaluated using 5-fold cross validation, split at the student level to evaluate how the model performs for unseen students. It is often common, in working with neural networks, to train using mini-batches of samples, updating model weights based on the outputs over several training steps. In the case of recurrent neural networks, the data contains multiple time steps that the model treats as a batch and updates the network weights at the end of the sequence. We update the model after each sample sequence using an adaptive gradient descent calculation [DHS11], and categorical cross-entropy is used as the cost function for model training due to its ability to handle multi-label classification; each sample contains a varying number of individual time steps, over which the network makes a single update from the aggregated cost.

Each model is trained over a multitude of epochs, or full cycles through the training set. Training over too many epochs or too few can reduce performance through overfitting and underfitting respectively. The appropriate number of epochs will also differ when applying models of different complexities, as is being done in this work. For this reason, we hold out 20% of each training set as a validation set and incorporate an "early stop" criterion for model training. After each epoch the model evaluates its performance on the unseen validation set to determine the point in training where there is little or no improvement.

A moving average of the model's error on the validation set, expressed as average cross-entropy (ACE) for training, is calculated over the most recent 10 epochs (starting with the 11th epoch). The model stops training when it finds that moving average value at a particular epoch is larger than or equal to the previously calculated average (lower values indicate superior ACE values). Using this criterion allows for a more fair comparison of the performance of each model. Although a

maximum number of 100 epochs was allowed, no models in this paper reached that maximum threshold.

3.4 Measures

We will evaluate the results of each of our model evaluations through three statistics, AUC ROC/A', Cohen's kappa, and Fleiss' kappa. Each kappa uses a 0.5 rounding threshold. This is a multi-label classification task such that each sample has one of four possible labels of confusion, concentration, boredom, or confusion. For this reason, the metrics of AUC and Cohen's kappa are first calculated for each of the four labels independently, and the final result is an average across the four labels [HT01]. It is not common to report average Cohen's kappa for multi-label classification; we include this metric for comparison to previous results reporting this metric. We also report Fleiss' kappa, which is better suited for multi-label classification, taking all label comparisons into account in a single metric. Both kappa metrics are reported as secondary measures, as AUC is unaffected by scaling and rounding threshold-setting procedures. In all cases, we report performance on the test data, averaged across each fold of a 5-fold cross validation.

3.5 Results

3.5.1 Adjusting the Dropout Context

Our initial analysis pertains to the degree of impact the context of dropout has on model goodness. We investigate this question in the context of the GRU model and the resampled training dataset, looking at whether dropout occurs before the recurrent layer, after the recurrent layer, or both. In all cases, a 30% hyperparameter

Model	AUC	Cohen's Kappa	Fleiss' Kappa
30% Dropout Before Recurrent Layer	0.74	0.12	0.22
30% Dropout After Recurrent Layer	0.74	0.13	0.23
30% Dropout Before & After Recurrent Layer	0.73	0.11	0.21

Table 3.1: Comparing locations of dropout within the GRU model.

is used for the dropout percentage. Table 3.1 shows that when dropout occurs has little impact on performance. When dropout is applied to both areas of the model, however, there is a mild reduction in both metrics, suggesting that applying dropout in both locations impedes model training to a noticeable degree. For this reason, all further models reported used dropout applied after the recurrent layer. This placement is chosen as there is a very slight increase in both Cohen's and Fleiss' kappa; additionally, it is more common for researchers and practitioners to apply dropout after the recurrent layer.

3.5.2 Comparing RNN Variants

We next compare a traditional recurrent neural network (RNN), a Gated Recurrent Unit (GRU) network, and a Long-Short Term Memory network (LSTM), which vary in their complexity, and as such in their number of parameters and flexibility of fit. These models are compared using the same training and test data sets and differ only in the internal node structure used for the network. In parallel, we examine the effects of adjusting the training data (but not the test data) using resampling, by comparing each model variant trained on the resampled dataset to that model variant trained on a data set without resampling.

The performance of each model is compared in Table 3.2. In all three model variants, training on the non-resampled data produced superior performance in all

Model	AUC	Cohen's Kappa	Fleiss' Kappa
RNN With Resampling	0.73	0.14	0.22
GRU With Resampling	0.74	0.13	0.23
LSTM With Resampling	0.73	0.11	0.22
RNN Without Resampling	**0.78**	0.19	0.24
GRU Without Resampling	0.77	0.19	0.24
LSTM Without Resampling	0.77	**0.21**	**0.27**
Wang et al. [WHH15]	0.66	0.25	–
Ocumpaugh et al. [OBG+14]	0.65	0.24	–

Table 3.2: Three recurrent model variants, trained on both the resampled and non-resampled datasets, are compared to the previous highest reported results on the ASSISTments dataset.

metrics over training with the resampled data, contrary to our initial hypothesis. Also contrary to our initial hypothesis, the GRU models did not produce the best outcomes; instead, the simplest model, the traditional RNN, was found to have superior AUC performance to the other models, albeit only by a small margin. This may be because it had the fewest parameters; the RNN trains approximately 82,000 parameters as compared to the over 244,000 parameters in the GRU model and nearly 326,000 parameters in the LSTM model. This smaller number of parameters also leads to the RNN being the fastest model to train. The LSTM model, however, had higher kappa values than the other network variants, and as such, could also be argued to be the best model as it exhibits comparably high AUC values and also would be able to handle longer sequences than a traditional RNN if used in real-time applications. All three deep learning models achieve substantially better AUC than the best models produced through prior work using more traditional machine learning algorithms (e.g. [OBG+14][WHH15]). Cohen's kappa, however, is found to be slightly worse than in the prior efforts.

Performance was generally good for AUC across all affective states, as shown in Table 3.3. It becomes apparent, however, that performance is not well-balanced across the labels. The difference between AUC and kappa values suggests that the model for confusion, for example, is generally able to distinguish between confused and non-confused students, but is poor at selecting a single threshold for this differentiation. The difference between affective states is likely associated with their relative frequency; the best-detected affective states (concentrating and boredom) were also the most common ones. While resampling was chosen to address this problem, Table 3.3 also shows that this technique, as implemented, did not lead to better performance.

3.6 Discussion and Future Work

Despite their broad application in other domains, deep learning models have been relatively under-utilized in education and their application often has not led to better results than other common algorithms [KLM16]. In this paper, we attempt to apply deep learning to the problem of sensor-free affect detection, using a data set previously studied using more traditional machine learning algorithms. Three deep learning models (RNN, GRU, and LSTM) were compared to previously published work. All three deep learning models explored here obtained substantially better AUC than past results reported using the same dataset, although they did not lead to better values of Kappa. This difference between metrics is not surprising, given that the cost function implemented in the deep learning models does not round each prediction before evaluating each class label, but instead evaluates the degree of error across all classes each training step. Nonetheless, the substantially higher AUC values argue that deep learning models may prove a very useful tool for research

	Resampled		Non-Resampled	
	AUC	Cohen's Kappa	AUC	Cohen's Kappa
Confused	0.67	-0.01	0.72	0.09
Concentrating	0.78	0.24	0.80	0.34
Bored	0.76	0.18	0.80	0.28
Frustrated	0.68	0.01	0.76	0.15
Average	0.73	0.11	0.77	0.21

Table 3.3: LSTM model performance for each individual affect label.

and practice in sensor-free affect detection, eventually leading to models that can be more effectively used both to promote basic discovery and to drive affect-sensitive intervention.

There are several aspects of the deep learning models that may have contributed to the improved AUC over the previous machine learning approach to constructing affect detectors for this dataset. In previous detectors, four separate models were built, trained, and evaluated independently while the deep learning model allows all four affective states to be evaluated and updated together with each training sample; such a process likely helps the model determine aspects of the data that help to make more accurate distinctions between each affective state in a temporal sense. Another aspect is in the flexibility of fit supplied by the neural network, allowing the model to capture the high complexity in student affect. This flexibility, however, also exhibits a drawback in terms of lacking interpretability; the large number of parameters and complexity of each model used in this work make it infeasible to study and understand how the model makes its predictions from the features it has available, particularly as it learns from previous time steps. At best, we can understand that the model is relatively better at predicting the more common categories (boredom and concentration) than the more scarce classes (frustration and confusion).

It is desirable to achieve excellent predictive accuracy for the more scarce, yet very important, affective states, in addition to the more common labels. It is possible that a different resampling approach could be more productive, although any resampling approach will be limited by the inter-connection of the observations, leading to non-uniformity across the labels; it is likely that in duplicating sequences containing the scarce labels numerous times, the model overfit to these sequences, which led to poorer extrapolation to unseen data. A possible alternate approach for the iterative refinement of these models would be to send field coders to classrooms working through material that is known to be more confusing and frustrating (e.g. [SBO+16]).

One further aspect not addressed by this work is differences introduced by student geographical factors. Earlier affect detectors in ASSISTments were found to perform relatively poorly on rural students when trained on urban and suburban populations [OBG+14]. Analyzing how robust deep learning models of affect are to population differences will help us to understand the degree to which these models generalize.

Chapter 4

Developing Early Detectors of Student Attrition and Wheel Spinning Using Deep Learning

Botelho, A. F., Varatharaj, A., Patikorn, T., Doherty, D., Adjei, S. A., & Beck, J. E. (2019). Developing Early Detectors of Student Attrition and Wheel Spinning Using Deep Learning. *Journal of IEEE Transactions on Learning Technologies.* (In Press)

Abstract

The increased usage of computer-based learning platforms and online tools in classrooms presents new opportunities to not only study the underlying constructs involved in the learning process, but also use this information to identify and aid struggling students. Many learning platforms, particularly those driving or supplementing instruction, are only able to provide aid to students who interact with the system. With this in mind, student persistence emerges as a prominent learning construct contributing to students success when learning new material. Conversely, high persistence is not always productive for

students, where additional practice does not help the student move toward a state of mastery of the material. In this paper, we apply a transfer learning methodology using deep learning and traditional modeling techniques to study high and low representations of unproductive persistence. We focus on two prominent problems in the fields of educational data mining and learner analytics representing low persistence, characterized as student "stopout," and unproductive high persistence, operationalized through student "wheel spinning," in an effort to better understand the relationship between these measures of unproductive persistence (i.e. stopout and wheel spinning) and develop early detectors of these behaviors. We find that models developed to detect each within and across-assignment stopout and wheel spinning are able to learn sets of features that generalize to predict the other. We further observe how these models perform at each learning opportunity within student assignments to identify when interventions may be deployed to best aid students who are likely to exhibit unproductive persistence.

4.1 Introduction

The use of digital learning environments in schools has led to new opportunities to study influential student learning constructs both longitudinally and at fine levels of granularity. Digital learning environments have emerged to take advantage of these opportunities, providing researchers with the tools and data to better understand such learning processes while simultaneously providing a platform through which that research can be implemented and deployed to improve students learning experiences. As is the case for many, if not all, learning platforms, particularly those that aim to drive or supplement teacher instruction, are only able to provide aid to students who interact with the system; it is for this same reason that human tutors

often employ a range of techniques to maintain student engagement and encourage student persistence when approaching difficult content [RC07]. This reinforces the need to better understand student persistence during the learning process so as to develop better detectors of struggling students and subsequently develop interventions to promote productive learning strategies.

When approaching difficult content, it is essential for students to exhibit high persistence by working through a sufficient number of practice problems in order to successfully learn the material. In this way, the construct of persistence plays an important role in student success as has been studied through research pertaining to grit [DPMK07], perseverance [PS+04], and productive failure [Kap08]. Students who fail to complete their work after only a small number of problems, defined in this paper as students exhibiting "stopout," are missing opportunities to learn difficult material through additional practice; this is particularly the case when students exhibit stopout early in an assignment, within, for example, the first few problems.

Although the presence of persistence is essential for students to overcome learning obstacles, there are cases where high persistence can be unproductive. This negative aspect of exhibiting high unproductive persistence has been operationalized in previous works through a behavior known as "wheel spinning" [BG13]. Wheel spinning describes the case when a student persists in a particular learning task yet is unable to reach a state of mastery within a reasonable timeframe.

Both stopout and wheel spinning represent unproductive examples of student persistence; in one case, stopout represents students who are not exhibiting enough persistence to succeed while wheel spinning represents too much persistence where it would likely benefit the student to stop and seek additional aid from an instructor or tutor. For this reason, we define stopout and wheel spinning as mutually exclusive measures within a single assignment. As previous works have defined wheel

spinning behavior as a student reaching the tenth problem, or learning opportunity, of a mastery-based assignment (discussed further in Section 3), students are only considered to have stopped out of an assignment if done before the tenth problem; it is important to emphasize this definition as each measure in this way represents what we consider to be unproductive learning behavior.

It is important to be able to detect when students are likely to exhibit stopout or wheel spinning behavior in order to develop interventions to promote persistence when it is likely beneficial to students and to also suggest additional help when such persistence is unlikely to lead to success. In light of this importance, however, deploying an intervention once stopout is detected is likely not very impactful as the student has already ceased interaction with the system, and similarly, in the case of wheel spinning, deploying an intervention at the moment of detection is likely too late as the student has already wasted time and effort (and perhaps has become frustrated). It is with these scenarios in mind that it becomes imperative to deploy such interventions preemptively in anticipation of such behavior and address potential causes of stopout and wheel spinning behavior before the student exhibits unproductive forms of high and low persistence. As will be discussed further in the Background Section, recent applications of deep learning in the context of education has led to promising results, supporting the exploration of such models for the task of developing early detectors of these student behaviors.

It is the goal of this work to explore the early detection of unproductive persistence as operationalized through wheel spinning and stopout. Using machine learning techniques including the application of deep learning in conjunction with both model and outcome transfer learning methods, we explore the relationship between learned predictors of wheel spinning and stopout both within an assignment and across assignments. With this goal in mind, we seek to address the following

research questions:

1. How do temporal deep learning models compare to traditional methods in the task of predicting wheel spinning and stopout behavior both within- and across-assignments?

2. Are learned predictors of each wheel spinning and stopout behavior also predictive of the other respective behavior (e.g. are predictors of wheel spinning also predictive of stopout as well as the reverse)?

3. How does recency affect the performance of models predicting each within and across assignment wheel spinning and stopout?

The focus of this work is on exploring the relationship between representations of unproductive student persistence in an effort to develop early detectors of such behaviors. The following section will first describe existing works that have previously studied behaviors of student attrition and wheel spinning in addition to previous applications of deep learning in the context of education. We will then describe the source and attributes of the data used in this work before detailing the applied methodology and analyses conducted to study these student behaviors. The results of these analyses will then be discussed with particular focus on the early detection of each within and across-assignment stopout and wheel spinning behaviors. Finally, we will discuss the potential future work, highlight the contributions of this work, and discuss final conclusions from the conducted analyses.

4.2 BACKGROUND

4.2.1 Wheel Spinning

Several previous works have explored and have attempted to model student wheel spinning behavior in several platforms including Cognitive tutor [MCS16] and ASSISTments [BG13][GB15], while other work has explored policies to help prevent wheel spinning [KKG16]. As described in the Introduction Section, wheel spinning is the behavior in which a student exhibits high persistence in a learning task, but unable to obtain sufficient understanding of the learning materials. The term "wheel spinning" is analogous to a car that is stuck in snow or mud; despite devoting effort into moving, the wheels will spin without getting anywhere.

In this work, we will be using the definition of wheel spinning given in the work of Beck and Gong [BG13] as failing to reach mastery after seeing ten learning opportunities. It is for this reason that prior work observing wheel spinning has pertained to student interactions with mastery-based assignments. Mastery-based assignments, as opposed to traditional assignments that require students to answer all assigned problems, instead require students to demonstrate a sufficient level of understanding, or mastery, of the assigned material in order to complete the assignment. In the case of ASSISTments, this threshold of understanding, by default, requires students to simply answer three consecutive problems correctly on the first attempt without the use of computer-provided aid.

Previous attempts to model wheel spinning have observed student activity on mastery-based assignments at the problem-level to predict whether the student will eventually wheel spin in that assignment [GB15]. The model was trained on expert-generated features describing each problem and student recent actions to estimate the likelihood of a student wheel spinning on the current assignment. We hypothe-

size that such a model is likely to perform better on later problems in an assignment than earlier problems, but previous works have reported an average model performance across all opportunities, or problems.

This paper attempts to, in part, build upon this previous body of work to build models to predict wheel spinning using a finer-granularity of data (e.g. at the action-level), observe wheel spinning behavior (as well as stopout which will be described next) over longer periods (e.g. across assignments), and observe how model performance changes over consecutive problems.

4.2.2 Student Attrition and Stopout

Student attrition, more commonly characterized by student dropout, has received a large amount of attention in recent years as a problem in education, largely due to its prominence in digital environments such as Massive Open Online Courses (MOOCs) [CRK15][XCSM16][YSAR13][RCY+14][LSHR15]. In such systems, it has been observed that a large portion of students do not complete their courses; such behavior is called dropout. Surveys have shown multiple reasoning behind low persistence in MOOCs which vary from learners to learners. For example, some may quit due to insufficient background knowledge or the difficulty of content, but other may get interrupted due to time management or scheduling, or simply stop coming back because they learned all they want to know [KH15]. Student attrition within MOOCs has also been previously studied through the development of a deep learning model, named "GritNet," that was found to outperform existing baseline methods [KVG18b] and even transfer across courses [KVG18a]. While these areas have, as described, received a large amount of attention, the characteristics of persistence and the reasoning for attrition in MOOCs differs greatly from that observed in K-12 classrooms as most students do not exhibit dropout in the same manner.

Dropout is not common within traditional K-12 classroom context (i.e., mandatory education) as attendance and graduation are often enforced and encouraged by the parents. Instead, student attrition and low persistence are observed in a form of students not completing certain learning tasks; we call this behavior "stopout". The main difference between stopout and dropout is that when a student stopouts, they are still in the course and may choose to complete the subsequent assignments, while learners are defined as dropout when the do not come back to finish the course.

When Student attrition at the assignment level, in many cases, prevents students from sufficiently learning the material and subsequently may lead to further difficulty when learning post-requisite skills (e.g. see [BWH15]), but also introduces a range of other issues pertaining to the development and deployment of effective learning interventions. As students exhibiting stopout behavior cease interaction with the learning environment, aid cannot be given to the student through the platform, relying solely then on external sources, such as the teacher, to help the student. Missing or incomplete student data caused by attrition makes it difficult to study the learning process (as no data can be recorded for students who are not interacting with the system), measure the effectiveness of interventions through randomized controlled trials [HHSK00], and, as the cause of stopout is often difficult to identify, develop effective interventions to support more productive persistence. For these reasons, it is important to build models to help identify students likely to exhibit stopout preemptively so that we can better understand the early signs of the behavior and develop interventions to prevent it.

4.2.3 Deep Learning in Educational Contexts

The use of deep learning methods in the context of education and learning analytics has led to a growing body of research focusing on better modeling student

behavior and performance. Within this domain, a large number of such works have begun to utilize recurrent neural networks (RNNs) [WZ89], for their ability to model complex temporal patterns of student behaviors. These models have shown great promise in recent works modeling student knowledge and short-term performance [PBH+15][KLM16][XZVIB16], predicting student graduation [KVG18b] and real-time performance [KVG18a] in MOOCs, detecting student affective state [BBH17], and predicting long-term outcomes [SBPH18][YLYY18].

Despite the often-reported high performance of these models as applied to their respective tasks in education, the large number of learned parameters and complex model structures often make them difficult to interpret. While this difficulty applies to the learned parameters of the model, this does not mean that the estimates produced by the models are similarly uninterpretable and can be utilized to explore student behavior over time at fine levels of granularity (e.g. see [BBOH18]). Something as simple as observing the estimates themselves, or even model performance, over time can lead to better insights into the modeled behaviors as well as when action may best be taken through intervention.

The high complexity of deep network structures allows the model to learn rich feature embeddings, either explicitly (e.g. [ZXZ+17]) or implicitly (e.g. [YLYY18]), that better describe the data to make better-informed model estimates. In this way, such models also support the application of transfer learning [Pra93] to better understand the relationship between outcomes of interest by providing the means to observe how learned features generalize across prediction tasks.

Feature Name	Description
Action Type	One-hot encoding of the action (attempt, help request, etc.)
Attempt Count	The number of attempts made up to the current action
Hint Count	The number of hints requested up to the current action
Problem Count	The number of problems seen up to the current action
Probability of Action	The probability of the current action given the problem
Probability of Action Given Action Count	The probability of the current action given both the problem and the number of actions taken in the problem
Probability of Response	When an attempt, the probability of a student answering with the specific response given the problem
Probability of Response Given Action Count	When an attempt, the probability of a student answering with the specific response given the problem and number of actions taken in the problem
Cumulative Log Likelihood of Response	The cumulative log likelihood of a student answering with the specific response on the problem
Normalized Time Taken	The amount of time since the last action, z-scored within action type and problem
Used Penultimate Hint	Whether the second-to-last hint has been seen before the current action
Used Bottom Out Hint	Whether the student has seen the last hint (containing the answer) before the current action
Correctness	Correctness or incorrectness if the current action is an attempt, or a non-attempt (as a 3-value one-hot encoding)
Preceding 3 Actions	One-hot encoding describing the previous three actions taken excluding the current action
Current and Preceding 2 Actions	One-hot encoding describing the previous three actions taken including the current action (current and previous 2)

Table 4.1: Description of the generated action-level features.

4.3 Dataset

The data used in this work is comprised of students working with ASSISTments during the 2016-2017 academic year. ASSISTments is a web-based learning platform that provides the tools for teachers to assign classwork or homework content for which students receive immediate correctness feedback [HH14]. While working through each assignment, many problems supply students with optional on-demand computer-provided aid; hints, of which there may be from 0 up to several available, supply students with an instructional message, while scaffolding, when available, breaks the problem into smaller steps to solve. In addition to these, the system

provides a "bottom-out" hint for every problem that supplies the students with the correct answer if the student is unable to solve the problem as students are not allowed to progress to subsequent problems until the correct response is entered inside ASSISTments.

ASSISTments is used by several thousands of distinct students daily, most of which being in 6th-8th grade solving primarily mathematics content, providing a dataset of sufficient scale and variation to apply deep learning methods that often require such data. While the majority of students are of late-middle-school age, the dataset itself is comprised of all users of the system during the aforementioned academic year. The data is filtered to include only student interaction with mastery-based assignments, known as "skill builders" in the system, where the completion threshold is designated to simply require students to answer three consecutive problems correctly without the use of computer-provided aid (i.e., without hints, scaffolding, or bottom-out hints). In recognition of wheel spinning as an undesirable learning behavior, the system implements a "daily limit," stopping students on the skill builder assignment for the day if the completion threshold is not reached by the tenth problem (except in the case where the student is about to reach the threshold on or directly following the tenth problem); the system provides the student with an instruction to seek additional help and return to the assignment on the subsequent day.

As teachers using the system assign a range of content, both made available through the system as well as self-built material, we include data from skill builder assignments where at least 10 students started the assignment and the overall completion rate is at least 70%. These limitations help to remove outliers such as sample classes and optional supplementary assignments where the teacher does not require every student to complete. These outlier cases are excluded as we would argue that

attrition due to such factors is not stopout as we have defined it within this task (e.g. low unproductive persistence).

4.3.1 Features

The data consists of action-level data recorded by the system, describing a fine-grained level of interaction with the content. As such, each row of the data describes a single action taken by a student pertaining to problem answering, or attempts, as well as hint requesting within the system in addition to time-related measures, probability of each response (e.g. identifying common wrong answers), and recency information (e.g. preceding actions taken). From the 15 features generated, a one-hot encoding was applied to all categorical features, resulting in a total of 86 features to use as input into our models. A brief description of each of these features is provided in Table 4.1.

4.3.2 Wheel Spinning and Stopout Labels

The labels of wheel spinning and stopout are applied to the data largely following previous definitions of these behaviors, although with a small number of edge-case exceptions that are detailed here to avoid ambiguity. As we hypothesize that wheel spinning and stopout are, respectively, representations of high and low unproductive persistence, as emphasized in the Introduction, we have defined these behaviors as mutually exclusive. Wheel spinning occurs when students have not reached a sufficient threshold of understanding by the tenth learning opportunity; we acknowledge that this threshold of ten problems to define wheel spinning behavior is rather arbitrary (and perhaps worth refinement in future work), but is used here for consistency with previous works studying wheel spinning behavior. Again as emphasized in the Introduction, we define stopout to occur only if a student fails to complete the as-

signment and stops out before the tenth problem. Attrition exhibited after the tenth problem is not labeled as stopout behavior, but rather would be characterized as wheel spinning (as the tenth problem was reached without completing the mastery assignment). In this way, any student with ten or more problems, unless completion was reached precisely on the tenth item, is labeled as having exhibited wheel spinning behavior.

The labels of each stopout and wheel spinning are represented as separate binary values and, while calculated at the student-assignment level, are applied to each row of the dataset. In this way, all models reported in this paper are predicting wheel spinning and stopout at each action taken by a student, similar to the problem-level estimates observed in previous works [BG13][GB15]. While we do not expect that such models will perform at the same level of accuracy for all actions, this level of prediction will allow for the study of such performance over time.

For this work, four labels are applied to the data corresponding to within and across-assignment indicators. In other words, a within-assignment wheel spinning and stopout (whether the student exhibits each behavior on the current assignment on which a student is working) is applied in addition to indicators of wheel spinning and stopout on the subsequent assignment. In both cases, a label is applied to each row of the data, again, corresponding to a single action taken by the student. In this way, next assignment wheel spinning and stopout behavior will be predicted from,

Number of Distinct Students	12,714
Number of Student Assignments	123,539
Number of Rows (Actions)	1,055,588
Percent Assignments with Wheel Spinning	4.85%
Percent Assignments with Stopout	4.72%

Table 4.2: The notable descriptives of the dataset.

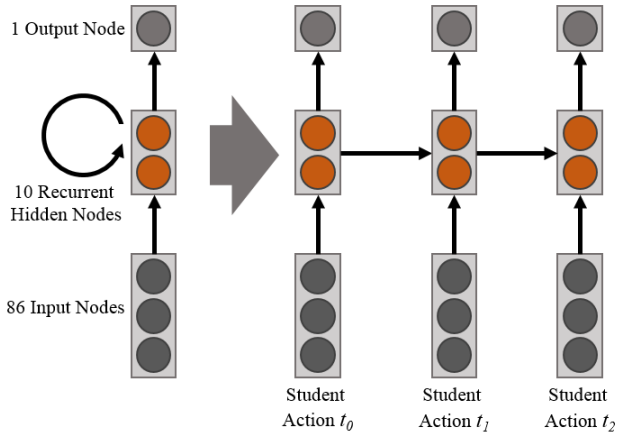

Figure 4.1: A simplified representation of the LSTM model structure, illustrating how information flows from previous timesteps to inform each model estimate.

for example, the first action of the previous assignment, then the second action, and so on. Similarly, as there is no included indication of the subject matter of the subsequent assignment, models of across-assignment representations of wheel spinning and stopout behavior is inherently capturing student-level (e.g. content agnostic) representations of such behavior.

The resulting dataset, as described by Table 4.2, contains over 100 thousand student assignments from over 12 thousands students, resulting in approximately 1 million actions to be used by our models.

4.4 Methodology

The methods used in this paper aim to address the research questions outlined in the Introduction Section centered on the application of a deep learning model in conjunction with transfer learning to predict both within and across-assignment representations of unproductive persistence. In this way, we develop a recurrent deep learning model as a means of learning a rich set of embedded features that are

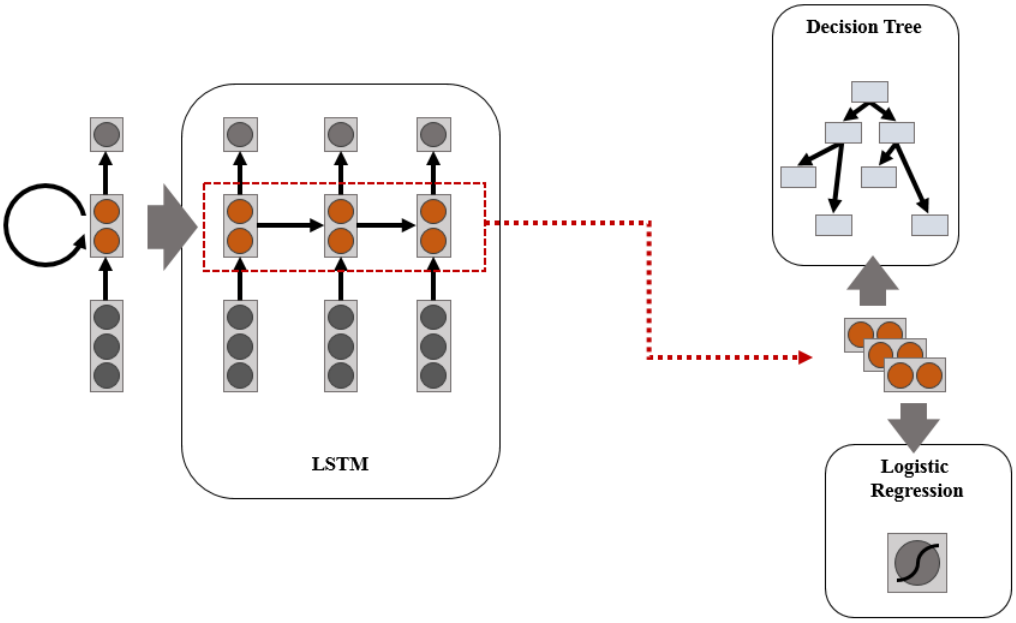

Figure 4.2: A visual example of the transfer learning procedure. The hidden layer of the trained LSTM model is used as input to train each a decision tree and logistic regression to predict each wheel spinning and stopout behavior.

predictive of one outcome (i.e., wheel spinning) in order to then observe how well such features generalize to predict the other outcome (i.e., stopout). This section will detail the models used to accomplish this goal as well as the set of methods applied in addressing our research questions outlined in the Introduction.

4.4.1 Building Models of Wheel Spinning and Stopout

In order to predict within and across-assignment wheel spinning and stopout behavior, we utilize a type of RNN called a Long-Short Term Memory (LSTM) network [HS97], in addition to a traditional decision tree model and logistic regression. Previous works focused on predicting wheel spinning behavior have utilized a logistic regression approach using a large set of engineered features [BG13][GB15]. While a set of engineered features are also utilized in this work, the previous models of wheel spinning have attempted to model at the problem level and included a larger set

of contextual features that describe prior performance on each knowledge component, or skill, in the assignment; the set of features we use here allow us to observe student-level representations of each behavior and future work can certainly expand on this to include more contextual, content-based features.

For each of the four labels applied to the dataset, a separate logistic regression, decision tree, and LSTM model is trained to predict the respective label. For all models trained in this work, we evaluate each using a stratified 10-fold student-level cross validation (utilizing the same folds in all models for fair comparisons). Given the large imbalance of stopout and wheel spinning labels (as most students do not exhibit such behavior per assignment), we stratified each fold by first clustering students based on the percentage of assignments in which each exhibited wheel spinning and stopout behavior, and then folding each cluster into 10 even folds.

In the case of the more traditional decision tree and logistic regression models, the raw features are presented as input to the model, with each action delivered as an independent training sample; again, the outcome is predicted at each action of the student within the system. The resulting performance of each model is then calculated across all samples within each fold and averaged across the 10 folds. The traditional models were implemented using the Scikit-Learn library [PVG+11] in Python using the default hyperparameters, with the exception of the max depth of the decision tree having been restricted to 3 levels to avoid potential overfitting; these settings were used for all logistic and decision tree models described in this work.

The LSTM model, however, as a temporal model, differs slightly in terms of how samples are presented to the model as input during the training procedure. In this case, samples are grouped by student assignment, with each sample representing a series of actions taken by a student within each assignment. The entire series of

assignment-actions are presented to the model and a series of estimates (of equal length to the input) is produced. In this way, the model is trained as a sequence-to-sequence model with a dynamic, yet finite, sequence length (as students completed a varying number of problems). The model attempts to learn temporal relationships within each student assignment to better inform its estimates, but still produces the same number of outputs as the traditional models. Similarly, as some of the features represent recent activity, the comparison of the models will help reveal aspects of these temporal relationships; comparing the LSTM and traditional models, for example, will reveal if utilizing longer-term student performance history lead to better model performance.

The LSTM model was developed using the Tensorflow library [AAB+15] in Python with a 3 layer structure; the input layer included 86 nodes corresponding with each of the available action-level features which then was fed into a hidden layer of 10 LSTM nodes and proceeded to an output layer of 1 output node to which a sigmoid activation function is applied. Minimal hyperparameter tuning was conducted for this network in an effort to reduce the chances of providing an unfair advantage to the model; for sake of reproducibility, the model used an Adam update function [KB14], cross entropy cost function, step size of 0.001, a batch size of 32, and used 20% of the training set as a validation set to determine when to cease model training.

4.4.2 Transfer Learning

Once each of the models is constructed and evaluated in predicting within and across-assignment wheel spinning and stopout behavior, we apply a transfer learning approach to study the relationship between such constructs. We have hypothesized that wheel spinning and stopout behavior are two extreme measures of unproductive

persistence. By employing the use of transfer learning, we can test this hypothesis, that the two measures are closely related, by observing how well predictors of one behavior transfer to predict the other behavior.

For this task, we utilize the LSTM model as the basis for the transfer learning method. As a recurrent network, the structure allows the model to learn a rich set of features that attempt to utilize complex temporal relationships in the data to make better-informed estimates at each time step; this rich set of features is stored in the network's hidden layer and, though not directly interpretable, this set of features is learned during the model training process. This development of embedded features is well-studied in other deep learning models, such as those utilized for image processing [KSH12][EBCV09]. The LSTM model, while not identifying lines and shapes as is found in image processing tasks, learns temporal features that help to distinguish between cases of positive and negative labels. The LSTM model is trained as a sequence-to-sequence model (i.e. many-to-many), allowing a set of features to be extracted for each time step and subsequently presented as input into a separate model; it is in this way that transfer occurs, where the LSTM learns a set of features in its hidden layer that are then transferred to another model that observes a different prediction task. For example, as there are 10 nodes in the hidden layer of the LSTM, the model learns 10 features from the preceding sequence of action-level features (see Table 4.1) that distinguish positive from negative labels of the dependent variable (i.e. either stopout or wheel spinning); the 10 features are then extracted for each timestep and used as input to either the decision tree or logistic regression model. A simplified representation of this process is illustrated in Figure 4.2. The logistic regression and decision tree models are then trained to predict either stopout or wheel spinning at each timestep (i.e. at each student action), using the features transferred from the LSTM model.

	DT		LR		LSTM	
Features	**AUC**	**RMSE**	**AUC**	**RMSE**	**AUC**	**RMSE**
Raw	0.847	0.327	0.511	0.437	0.887	0.313
LSTM - Wheel Spinning	0.87	0.318	0.887	0.313	—-	—-
LSTM - Stopout	0.679	0.388	0.708	0.39	—-	—-

Majority Class Model RMSE: 0.482

Table 4.3: Predicting Wheel Spinning in current assignment

	DT		LR		LSTM	
Features	**AUC**	**RMSE**	**AUC**	**RMSE**	**AUC**	**RMSE**
Raw	0.706	0.224	0.46	0.275	0.759	0.223
LSTM - Wheel Spinning	0.71	0.224	0.683	0.226	—-	—-
LSTM - Stopout	0.747	0.223	0.757	0.222	—-	—-

Majority Class Model RMSE: 0.234

Table 4.4: Predicting Stopout in current assignment

With this methodology, four sets of transfer learning models are compared for each within and across-assignment labels of wheel spinning and stopout. These four sets compare different combinations of features, gained by training the LSTM model to predict either wheel spinning or stopout behavior, and each outcome. First, the features learned by the LSTM model to predict within assignment wheel spinning, referred to henceforth as the "wheel spinning features," are presented to a decision tree model and a logistic regression to predict within assignment wheel spinning; this task allows us to identify first any potential differences to performance caused by model transfer (it is not guaranteed that the subsequent model will be able to effectively learn how to utilize the features as the output layer of the LSTM had). Secondly, the wheel spinning features are again presented to a different decision tree and logistic regression model which are then trained to predict within-assignment stopout. The third set of models then observes, conversely, how well the stopout features, learned by the LSTM model trained to predict within assignment stopout,

transfer to a decision tree and logistic regression model to again predict within assignment stopout. Finally, the fourth set of models uses the stopout features in a decision tree and logistic regression to predict wheel spinning. It is important to clarify that this work does not attempt to make the comparisons between within-assignment features transferring to predict next assignment outcomes.

4.5 Results

4.5.1 Metrics

We compare the results using two primary metrics of AUC and RMSE in addition to, in the case of observing model performance over time, Recall. There are several benefits to using this particular range of measures to evaluate each model, particularly in case of modeling wheel spinning and stopout where there is a large imbalance amongst the labels (most students do not exhibit such behaviors). In such cases of imbalance, majority class models tend to appear to perform well even when no distinction between classes is learned. To prevent trained models from producing a low error by biasing their estimates toward majority class, we use AUC to evaluate model fit.

The use of AUC evaluates how well a model distinguishes positive samples from negative samples; given an instance of the positive class and the negative class, AUC can be thought of as the probability the positive class will be the one with a higher probability estimate. Therefore, the measure accounts for sparseness of the positive class. The value is bounded between 0 and 1, with higher values indicating better model fit. Values close to 0.5 are indicative of the model performing similar to random chance.

While AUC evaluates how well the model is able to distinguish the classes, RMSE

	DT		LR		LSTM	
Features	**AUC**	**RMSE**	**AUC**	**RMSE**	**AUC**	**RMSE**
Raw	0.581	0.238	0.539	0.273	0.600	0.251
LSTM - Next Assignment Wheel Spinning	0.595	0.250	0.601	0.250	—	—
LSTM - Next Assignment Stopout	0.570	0.251	0.569	0.251	—	—

Majority Class Model RMSE: 0.246

Table 4.5: Predicting Wheel Spinning in next assignment

	DT		LR		LSTM	
Features	**AUC**	**RMSE**	**AUC**	**RMSE**	**AUC**	**RMSE**
Raw	0.545	0.209	0.492	0.25	0.557	0.221
LSTM - Next Assignment Wheel Spinning	0.547	0.221	0.548	0.221	—	—
LSTM - Next Assignment Stopout	0.553	0.221	0.557	0.221	—	—

Majority Class Model RMSE: 0.215

Table 4.6: Predicting Stopout in next assignment

identifies the distance of each estimate (in terms of error) from the true label; the metric is calculated using the continuous-valued probability of each class as produced by the model and comparing this against the ground truth label. In this way, the model penalizes for indecisiveness in the model. For example, if, for a set of positive and negative labels the model produced all estimates of 0.1 and 0.09 respectively, the AUC would indicate perfect model fit while the RMSE would be comparatively poor (as the error on the positive instances is very high). This metric, however, does not account for majority class bias and should therefore be compared in relation to the RMSE value of a majority class model. The value of RMSE is bounded between 0 and 1 in this case (as all estimates are bounded within this range and the labels are binary values), with lower values indicating better model performance.

Finally, we will also report a value of recall when observing the next assignment wheel spinning and next assignment stopout models performance over time. Recall, as a measure of accuracy in regard to the positive label (for all positive cases, how many did the model successfully identify), helps to identify model performance in identifying the positive cases of wheel spinning and stopout. This is particularly

important, again, due to the large imbalance as it provides a means of evaluating the models ability to identify cases of stopout and wheel spinning behavior. The drawback of this metric is that it does require a rounding threshold to be set, and as it is likely the estimates are biased toward the majority class, a rounding threshold of the model output mean is used rather than the more traditional use of 0.5; in other words, values above the mean are rounded up to identify a positive case of either wheel spinning or stopout and estimates below the mean are rounded down to identify a negative case of either measure. The value of recall is also bounded between 0 and 1 with higher values indicating better model performance.

4.5.2 Model Performance

Our results are recorded such that each of the Tables 4.3-4.6 record results of one outcome variable. Table 4.3 describes the various models which where built to predict if a student is going to wheel spin in the current assignment. The first model was built using the raw features (i.e the original features of the dataset as listed in Table 4.1). We see that the LSTM model performs the best with an AUC of 0.887 and an RMSE of 0.313. It is then followed by the decision tree model with an AUC of 0.847 and RMSE of 0.327. The logistic regression model does not perform well with a low AUC of 0.511, barely better than chance performance.

The second and third model in Table 4.3 is built using transfer learning, where we use the learned hidden layer of the LSTM trained to predict wheel spinning. Its learned features are used as input to the decision tree and logistic regression models. This experiment demonstrates how well the learned features transfer between models as well as generalize to new outcomes. The main result is that both models show improvement when trained using the features discovered by the LSTM: decision trees see a slight improvement in both AUC and RMSE, while logistic regression

is greatly improved. We see that the LSTM-Logistic Regression model with AUC of 0.887 (RMSE 0.313) performs better than the LSTM-Decision tree model with AUC of 0.87 and RMSE 0.318. We can observe that the transfer of the LSTM model over the Logistic Regression model is resulting in the same AUC of the LSTM with raw features, which is unsurprising as the output layer of the LSTM is essentially a logistic regression model. The last model is another transfer learning model, wherein the LSTM which was built to predict stopout in the current assignment is used to transfer its learned features to a decision tree and a logistic regression model to predict wheel spinning. The results were mixed, with the decision tree exhibiting little benefit vs. using the raw features, while logistic regression outperformed the raw features. It is interesting that the logistic regression improved even when given features extracted for a different learning activity. We observe that both of these models performed well with an AUC of 0.679 and RMSE of 0.388 in the case of the LSTM-decision tree model and an AUC of 0.708 and RMSE 0.39 for the LSTM-logistic regression model.

Similar to Table 4.3, Table 4.4 records the model performance for predicting if a student is going to stopout in the current assignment. The order is similar to Table 4.3 where in the first row the original features were used to fit the decision tree, logistic regression and the LSTM model. The LSTM model seems to perform the best with an AUC of 0.759 and RMSE of 0.223, followed by the decision tree and logistic regression models with AUCs of 0.706 and 0.46, respectively. The second model is the first of the transfer learning models aimed at predicting within-assignment stopout. The learned features of the LSTM to predict wheel spinning were transferred as input to a decision tree and logistic regression model to predict the stopout. Despite using features learned for a different prediction task, both the decision tree and logistic regression showed improved performance over just using

the raw features. For decision trees, the benefit is slight with a trivial increase in AUC. However, logistic regression demonstrated a large performance gain with AUC improving from 0.46 to 0.683 and RMSE improving from 0.275 to 0.226. Finally using the LSTM model which was built to predict stopout using the raw features, we transferred its learned features as input to a decision tree and logistic regression model to predict the same stopout label. Both models show improvement over using the LSTM stopout features. Both decision tree and logistic regression show noticeable performance gains in AUC, with smaller gains in RMSE.

Table 4.5 describes the results for the model built to predict if a student is going to wheel spin in the *next*, rather than current, assignment. Using the original features, the LSTM exhibits an AUC of 0.600 (RMSE 0.251), followed by the decision tree with an AUC of 0.581 (RMSE 0.238), and then the logistic regression with an AUC of 0.539 (RMSE 0.273). The second row describes the performance of the transfer learning models from the LSTM built to predict wheel spinning in next assignment. This LSTM - decision tree model had an AUC of 0.595 (RMSE 0.250) while the LSTM - logistic regression model exhibited an AUC of 0.601 (RMSE 0.250). Similarly the LSTM built to predict next assignment stopout is used to build transfer learning models with the decision tree and logistic regression for the task of predicting next assignment wheel spinning. These models resulted in an AUC of 0.570 (RMSE 0.251) for the transferred decision tree model and an AUC of 0.569 (RMSE 0.251) for the logistic regression model. Again, we observe the general pattern of learned features resulting in better accuracy than the raw features. For logistic regression, even features built for a stopout manage to outperform the raw features, although this result does not hold for the decision tree.

Table 4.6 describes the models built to predict if a student is going to stopout in the next assignment. Following the similar structure of the previous tables, the

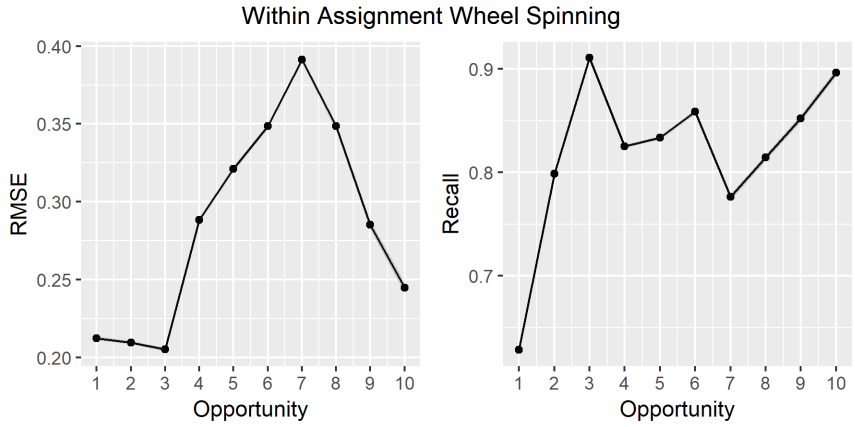

Figure 4.3: The performance of the LSTM model in predicting within-assignment wheel spinning by opportunity.

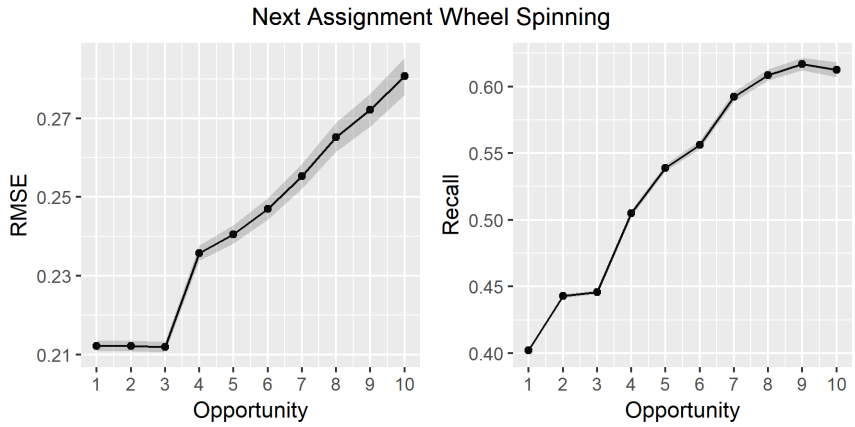

Figure 4.4: The performance of the LSTM model in predicting next assignment wheel spinning by opportunity.

original features were used to build a decision tree, logistic regression model and an LSTM model. The results are not nearly as strong as shorter-term predictions for the current assignment, but are still better than chance, perhaps highlighting the difficulty of identifying this behavior as early as the previous assignment without contextual information as to the content of the subsequent assignment. The LSTM again seemed to perform the best out of the three with a not-so-high AUC of 0.557 (RMSE 0.221). It was followed by the decision tree with a AUC of 0.545 (RMSE 0.209) and the logistic regression model with a below chance AUC of 0.492 (RMSE 0.25). Following the raw features, we use what was learned by the LSTM model built to predict wheel spinning in the next assignment to transfer its learning to a decision tree and a logistic regression model. These models resulted with AUC of 0.547 (RMSE 0.221) and 0.548 (RMSE 0.221), respectively. We observe that there are few differences between the two models. Next we use the LSTM model trained to predict next assignment stopout to transfer its learning to a decision tree and logistic regression model to predict the very same label of next assignment stopout, resulting in AUCs of 0.553 (RMSE 0.221) and 0.557 (RMSE 0.221) respectively.

It is important to reiterate that each model is predicting the respective label at each timestep. In other words, each behavior is predicted at each student action. It is likely for this reason that some models exhibit AUC values near chance; the poor performance of the logistic regression model in Table 4.3, for example, and conversely high performance of the decision tree, suggests that positive and negative labels of the behavior are not linearly separable using the raw features alone and need more information (such as the temporal features supplied by the LSTM) in order to exhibit higher performance.

4.5.3 Observing Model Performance by Opportunity

In addition to observing model performance averaged over all estimates, we further observe how model performance changes at each learning opportunity, or problem, when predicting each outcome measure. By observing how these models perform at each learning opportunity, we can begin to identify how early in the preceding assignment we are likely able to detect indicators of unproductive persistence in the future; this can then help to 1) identify potential causes or factors that may correlate with future unproductive persistence and 2) begin to understand not only when but also what type of intervention may be deployed to support productive learning behaviors.

As the data is represented as a series of student actions, we first take the mean model performance within each student problem and plot this performance over the first ten problems of the student assignments as shown in Figure 4.3. As the number of students present at each opportunity changes due to students either exhibiting stopout behavior or effectively completing the assignment, it is important also to include confidence intervals as each value will be less precisely measured at each subsequent opportunity. In the case of RMSE, this confidence interval is calculated by computing the square root of the upper and lower bounds of the standard errors calculated from the squared errors across estimates at each opportunity. In the case of recall, the confidence bounds are computed using a Wilson score interval [Wil27] for the computed recall value at each opportunity. The confidence bounds for AUC is computed using pROC [RTH+11], an an open source R package.

We plot the model performance for each within next assignmen t wheel spinning and next assignment stopout as estimated using the LSTM model without transfer learning in Figures 4.4 and 4.6 respectively; we compare these, then to the model performance for each within-assignment wheel spinning and stopout depicted in

Figures 4.3 and 4.5 respectively. It is important to highlight, as was described in the Metrics Section, lower RMSE values indicate better model performance while both higher recall and higher AUC values are indicative of better model performance; in this way, although both RMSE and recall, for example, exhibit a general upward trend over each subsequent opportunity, the metrics are contradictory in their trend of model performance. This particular case observed in Figure 4.4 would therefore suggest that, while the model is able to correctly identify a larger number of students likely to wheel spin by the end of the preceding assignment, the model is less precise in its ability to do so. This is further supported by the decrease in AUC observed in that figure, where the model is likely mislabeling students who do not wheel spin on the next assignment.

When predicting next assignment wheel spinning, as illustrated in Figure 4.4, the RMSE of the model is at its lowest over the first three opportunities of students assignments. This is not very surprising as, since the completion threshold for the assignments is answering three consecutive problems correctly, a large number of students will likely answer the first three problems correctly and effectively complete the assignment. Such students, although certainly dependent on content, are probably less likely to exhibit wheel spinning in future assignments than students exhibiting difficulty early in the assignment; students who do not effectively learn the material are likely to struggle to learn subsequent skills that may require mastery of the prior content. The model performance, in terms of RMSE, then steadily declines after the third opportunity as it is likely biasing estimates toward the majority class. In regard to both recall and AUC, however, the model is steadily improving with each subsequent opportunity, suggesting that, while perhaps biased toward majority class, the model is able to more effectively identify future cases of wheel spinning behavior as students remain in the assignment. The model's recall

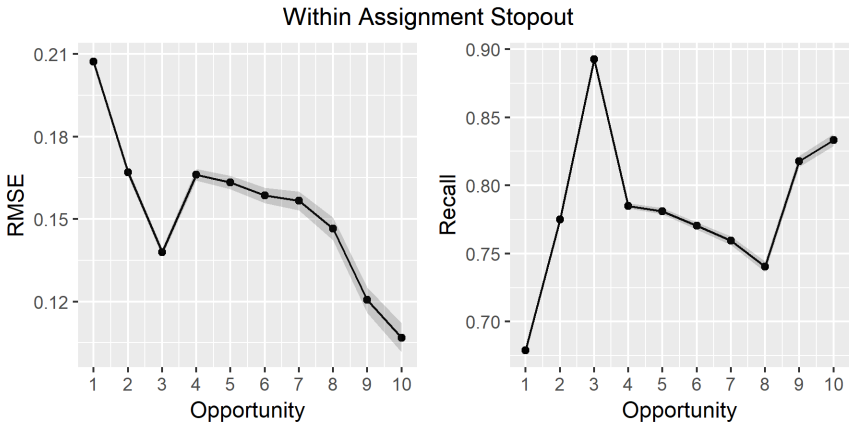

Figure 4.5: The performance of the LSTM model in predicting within-assignment stopout by opportunity.

does seem to plateau near the end of the 10 problem span, but the result suggests that by the end of the assignment, it is able to identify 60% of the wheel spinning students on the subsequent assignment (without even knowing what that content will be). Presumably, the model may be simply identifying cases where students who exhibit wheel spinning within the current assignment are more likely to wheel spin on subsequent assignments, particularly as the students remaining in the assignment at the tenth opportunity are wheel spinning (unless completion is reached on the tenth item per our definition of the behavior).

In one sense, this suggests that, somewhat unsurprisingly, an intervention aimed at preventing wheel spinning on a subsequent assignment is likely to be most impactful at the first sign of potential wheel spinning behavior on the current assignment. In the case of our results, this seems to be around the third learning opportunity, as illustrated by the recall and metric in Figure 4.3. In that figure, the third opportunity exhibits both the highest recall, suggesting that the model is able to identify the cases where wheel spinning is exhibited by the end of the assignment, and the lowest RMSE, which, even with majority class bias, is the opportunity where all metrics generally agree in terms of exhibiting good model performance.

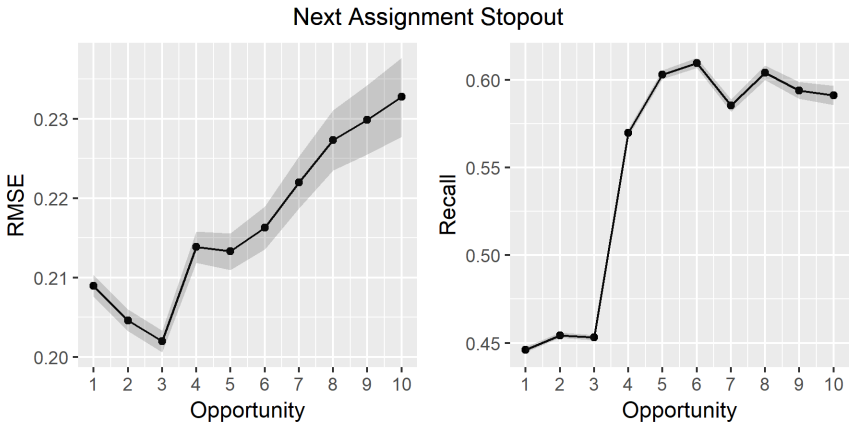

Figure 4.6: The performance of the LSTM model in predicting next assignment stopout by opportunity.

The performance of the LSTM model in predicting next assignment stopout, as depicted in Figure 4.6, illustrates a similar trend to that of the wheel spinning model. Although exhibiting noticeably higher variation, the RMSE of the stopout model is lowest within the first three learning opportunities and steadily increases on subsequent opportunities. Recall again exhibits a contradicting trend, exhibiting the worst performance over the first three opportunities and then substantially increasing in performance after the third opportunity, correctly identifying approximately 59% of the students who stopout on the next assignment. By the large confidence bounds on AUC, however, it would appear that, similar to the AUC of the next assignment wheel spinning model illustrated in Figure 4.4, the model has difficulty distinguishing students likely to exhibit each of these behaviors in the future.

In observing the within-assignment performance of this model in Figure 4.5, however, another interesting trend can be identified. Similar to the wheel spinning model, the metrics appear to agree in terms of better model performance on the third opportunity. However, the RMSE steadily improves and both the recall and AUC metrics decrease somewhat steadily after this point. This almost-inverse trend from what was seen for the wheel spinning performance suggests, although not

surprisingly, that the model is unable to distinguish students likely to stopout and persist on later opportunities; by our definition, stopout can only occur within the first ten opportunities, but also students present on later opportunities are demonstrating persistence which may be hard for the model to identify when stopout will occur in such cases.

4.6 Future Work

Although this work advances the understanding of transfer learning in understanding educational performance, there are several interesting followup questions. First, we found a general pattern of logistic regression benefiting from transfer learning, while the results for decision trees were more mixed. Is this trend a general one, or is it particular to our data set and set of features? Similarly, how would other classifiers such as random forests or decision stumps perform? Would they benefit from the constructed features or not? The first step here of exploring transfer learning is useful, but the field needs a better understanding of under what circumstances features will transfer to new learners.

The second area of investigation centers around the differing benefits transfer learners gain. When the features aligned with the task, e.g. stopout features for predicting stopout, both decision trees and logistic regression showed benefit. However, when the features were less aligned, such as wheel spinning features being used to predict stopout, results were more mixed. There are several next questions to ask in this area. First, how broadly applicable are the learnt feature sets? Would they show improvement over raw features predicting less-related tasks should as learner affect? Second, is it feasible to train a neural network with multiple outputs to encourage it to learn features that are more broadly applicable (e.g. through

multi-task learning [Car97])? In this way, a major area of research could be training networks on a variety of outputs and using the learnt features for a variety of novel research topics. Removing humans from feature generation may result in less interpretable features, but might result in both more accurate models and novel features we have not yet hand-discovered.

The final area we think worth pursuing is understanding the large dropoff in performance from predicting current problem set wheel spinning and stopout, to predicting next problem set wheel spinning and stopout. Some of the decrease in performance is fundamental to any prediction task: predictions further in the future have more uncertainty than about near-term events. How much of the decrease is a fundamental limitation, and how much is due to their not being as much prior art in longer-term predictions? Is it possible to increase accuracy on later problem sets to an AUC of 0.7 with better feature construction or model choices, or are there fundamental limits to how accurately we can predict student performance?

4.7 Contributions and Conclusions

This paper makes two contributions with regards to transfer learning. First, we have found that in some instances transfer learning works better than the original features. We were surprised that machine-learnt features, designed to work with a neural network, were applicable to a decision tree. Given the identical model forms, it was less surprising the features improved performance of logistic regression models. The second contribution is that transfer learning (sometimes) works for non-identical tasks. Using LSTM-stopout features for predicting wheel spinning, and vice versa, performance improved for the logistic regression models and sometimes improved for the decision tree models. This finding demonstrates that it is possible

to automatically construct features that are applicable to new prediction tasks.

This paper also makes contributions with respect to predicting longer-term events. Earlier work on student modeling focused on immediate events such as predicting how the student would perform on the current problem. Later work lengthened the prediction interval to see how a student would perform on a problem set, which was composed of many problems. This work increases the temporal interval to predict how a student will perform on the next problem set. In many ways, this work is a greater increase than going from current problem to current problem set, as in both cases the predictive model has information of how the student is performing on this skill. For predicting the next problem set, the model is unsure how the student will perform on the skill. Thus the predictive task is comparably more difficult.

In conclusion, this paper focuses on providing an early warning to predict which students will struggle. Providing help and additional learning resources to students who are struggling to learn is an integral part of any learning system. Identifying students who are going to struggle is crucial for helping these students; the sooner we know if a student is going to wheel spin or stopout, the better we can provide the right kind of help to the students. Prevention is better than cure, likewise it is better to prevent the student from wheel spinning or stopout than providing them with remedies later on. From our results, we can say that our models are good at identifying the stopout and wheel spinning behavior early from the actions of the students in the current assignment. From our models we can understand student persistence in the form of wheel spinning and stopout. Using these concepts, we can try to make students persist longer if they are not persisting long enough. Or we could stop them from persisting if we identify that they have been struggling for a long time. We can use these models to provide intervention at an early stage of the assignment such as when the model detects the behavior after an action made by

the student. If the model predicts if the student is going to wheel spin, we could stop providing the student with more problems for the day. Instead, we could point the student to a learning resource such as class notes or video. Similarly, if the model predicts if a student is going to stopout, we could try to lower the difficultly of the problems so that the student gains confidence in solving problems instead of stopping out. By using the detectors for next assignment behaviors, we are detecting vulnerable students an assignment early.

Chapter 5

Machine-Learned or Expert-Engineered Features? Exploring Feature Engineering Methods in Detectors of Student Behavior and Affect

Botelho, A.F., Baker, R.S., & Heffernan, N.T. (2019). *Machine-Learned or Expert-Engineered Features? Exploring Feature Engineering Methods in Detectors of Student Behavior and Affect.* (In Submission).

Abstract

There has been a long history of research on the development of models to detect and to study student behavior and affect during learning activities. The development of these models within computer-based systems has allowed the study of learning constructs at not only fine levels of granularity, but

also at scale by leveraging the large sums of student log data recorded by such systems. For many years, these models, regardless of their outcome measure, were developed using carefully engineered features based on previous educational research from the raw log data. More recently, however, the application of deep learning models has often skipped this feature-engineering step by allowing the algorithm to learn often-uninterpretable features from the fine-grained raw log data. As many of these deep learning models have led to promising results, the question has been raised as to which situations may lead to machine-learned features performing better than expert-generated features. This work aims to address this question by comparing the use of machine-learned and expert-engineered features for three previously-developed models of student affect, off-task behavior, and gaming the system. In addition to this comparison, we propose a third feature-engineering method that combines expert features with machine learning, to further explore the strengths and weaknesses of each of these approaches for use in building detectors of student affect and unproductive behaviors.

5.1 Introduction

The educational data mining community has developed numerous models to detect unproductive student behaviors and affective states and study how these measures correlate with short- and long-term learning outcomes. Estimates produced by detectors of student affective states and unproductive behavior, for example, have been found to predict student standardized test scores [PBSP+13], whether a student chooses to attend college [PBBH13], and whether they pursue a degree in STEM [SPOBH14], and even later pursue a STEM career [MM18], from estimates produced from interaction logs collected as they worked on mathematics problems

in seventh grade. The predictive power of these detectors along with a general desire to understand and improve the student learning process has led to a significant amount of research around developing these models.

For many years, these detectors, exploring a range of variables including that of student affect, off-task behavior, and gaming the system, have been built using sets of hand-crafted features based on prior education research. More recently, however, the application of deep learning models to raw data have shown promising results (e.g. [ZXZ+17][PHM+18]; such models often skip the task of feature-engineering by allowing the model to learn sets of embedded features using a machine learning approach rather than constructing features by hand. This has raised the question as to whether the often-arduous task of generating features by hand leads to more accurate models, or do features that are automatically distilled by a deep learning model lead to higher performing models?

5.1.1 Expert-Engineered vs. Automatically Distilled Features

The comparison of different sources of features, whether generated through a research-based engineering process or by means of a machine learning model, must consider a number of dimensions as each type of method provides certain affordances that may be desirable under different applications. It is similarly important to compare models utilizing these different sources of features, using several metrics to highlight particular strengths and weaknesses of each approach. Prior work, for example, suggests that machine learned features lead to better performance on some metrics, but worse performance on others [BP18][BBH17]. Also, additional attributes of models such as interpretability and ease of deployment should be considered to determine which approach is best overall for a specific application.

The primary goal of the current paper is to compare the two aforementioned methods of generating features to be utilized by models of student affect and unproductive behaviors. However, it is also important to consider whether some types of models may perform better for certain types of features. In other words, the choice of model may largely impact the benefit of different features or even restrict which types of features are possible at all. RNN models are able to easily accept sequences of labeled and unlabeled low-level action data for training. Conversely, other simpler models such as a decision tree or logistic regression would require some type of feature engineering, or aggregation, in order to incorporate labeled and unlabeled data; additionally these models would be unable to easily observe unlabeled data in a semi-supervised learning manner, often referred to as co-training, as can be accomplished with a recurrent deep learning network [WZ89]. It is important to note that we use the term "co-training" to describe the use of both labeled and unlabeled data to inform model estimates and the methods differ from that of other works describing co-trained models [BM98]. It is also the case that recurrent deep learning networks are not the only manner in which co-training can be performed (e.g. [AG02], [MRS02]) but this is the method used in the analyses described in this paper. Whether or not the use of unlabeled data makes a difference in regard to model performance is a different question - one that will be addressed in this work - but it is difficult to fairly compare the benefits of methods of feature engineering without also considering the types of models that utilize such features.

Commonly, as is the case in this work, the generation of features through machine learning methods refers to the use of a deep learning model, as the complex structure is often believed to learn sets of features within a number of hidden layers; this is perhaps best exemplified in image processing domains where the features learned by certain types of deep learning models can be extracted and visually inspected

[MMCS11]. In non-image data, such as the student interaction logs observed in this work, it is difficult if not impossible to interpret the features learned by such a model, particularly when applying recurrent neural networks (RNNs) [WZ89] that attempt to learn temporal or sequential relationships within a set of data. The lack of interpretability of these deep learning models detracts from their utility in research settings as it is difficult to justify why the model produced a particular estimate; when one cares about the importance of features in a prediction model, the use of these deep learning models offers little benefit. Despite the ambiguity of these models, they can perhaps be useful in some in educational contexts, as previous work has effectively used deep learning models to learn temporal trends among and across affective states [BBOH18]; this was accomplished by studying each affective state based on the output estimates of the model without the need to interpret any learned parameters within the model.

5.1.2 Research Questions

As has been described, the goal of the current work is to compare the strengths and weaknesses of differing feature engineering methodologies through both the performance of models utilizing such features according to multiple metrics and also regarding interpretability and applicability. Specifically, we re-develop detectors of student affective state [BBH17], off-task behavior [PBSP+13], and gaming the system [PBdCO15], comparing new models to previously-developed models, to address the following research questions:

1. Which leads to better model performance (AUC ROC and Kappa), expert-engineered features or machine-learned features, for detectors of affect and unproductive behaviors?

2. Does the combination of expert-engineered features and machine learning-based feature generation lead to any improvement in model performance for detectors of affect and unproductive behaviors?

3. Does the incorporation of unlabeled data through model co-training lead to any improvement in model performance for detectors of affect and unproductive behaviors?

5.2 Background

The comparison of expert-engineered features and those generated through the use of a machine learning model has been conducted previously on similar detectors within the computer-based learning system known as Betty's Brain [JBB+18]. In that work, only small differences were found between models using expert-engineered features and models utilizing features automatically distilled through the use of a deep learning model. While the comparisons made in that work are arguably inconclusive, it raises many of the questions posed in this current work. The inconclusive findings of [JBB+18] motivates a need to understand which contexts one method of feature generation may be better over another in developing accurate detectors of student affect and disengaged behavior.

Detectors of student affective state have been developed in a number of learning systems including Cognitive Tutor [dBGW+12], AutoTutor [DCW+08], Crystal Island [SML11], MathSpring [HWBA18], Betty's Brain [JBB+18], and ASSISTments [OBG+14][WHH15], the last of which supplied the data used in this current work. While some projects have sought to develop these detectors with the help of physical and physiological sensors [DLS+10][ACB+09][PRB+16], we instead focus on the development and application of sensor-free detectors of student affect as well as dis-

engaged behavior. In such detectors, each label is inferred using only interaction logs collected through a particular learning platform.

The development of affect detectors within ASSISTments has undergone several iterations of improvements. From some of the initial work exploring the use of expert-engineered features to develop and evaluate detectors through a population validity study [OBG+14], additional feature engineering work focused on improving the skill-based features by exploring the knowledge components associated with problems within the dataset [WHH15]. More recently, a deep learning approach was applied [BBH17], utilizing the expert-engineered features within a recurrent neural network to predict the four labels of affective state (i.e. engaged concentration, boredom, confusion, and frustration) simultaneously over time; it is this model that is used for comparison in the current work.

Student off-task behavior has also been studied in a number of systems including Cognitive Tutor [AMRK06], CIspace [AC06], and ASSISTments [PBSP+13]. This behavior is often characterized by such behaviors as talking to other students or engaging in tasks unrelated to assigned work [BCKW04].

Detectors of gaming the system have similarly been previously developed in a number of learning systems including Cognitive Tutor [Ale01][BCKW04], Reading Tutor [Jos05], Wayang Outpost [JW06], as well as ASSISTments [PBdCO15]. Previous work on such detectors on data collected within Cognitive Tutor explored a number of features found to be predictive of student gaming behavior [PdCBO14], and then later studied how such features generalize between learning systems [PBdCO15], leading to the detector model observed in this work.

The three previously-developed models of student affective state, off-task behavior, and gaming the system represent, to the authors' knowledge, the highest performing detectors of their respective outcome previously published using AS-

SISTments data (the data itself will be described in greater detail in the next section).

5.3 Data and Labels

This work utilizes two datasets consisting of student interaction log data collected within the ASSISTments computer-based learning platform. The content within the system consists primarily of mathematics problems, as is the data used in this work, for students in grades 6-8. While the system itself is not limited to mathematics and contains content from early elementary school through early-college, the majority of teachers and students use the system for middle school mathematics homework and classwork. Students working in the system receive immediate correctness feedback on each problem with the ability to make multiple attempts, and have the ability to ask for on-demand computer-provided aid in the form of hints or scaffolded problems. These interactions and timing information are the data used to construct most of the expert-engineered features utilized in this work.

Each of the datasets is drawn from data utilized in previous published work to develop and study models of student affect [OBG+14][WHH15][BBH17][BBOH18], off-task behavior [PBSP+13], and gaming the system [PBdCO15]; the first dataset contains data pertaining to both student affect and off-task behavior labels, as these were collected in-tandem, while the second dataset contains data collected to study student gaming the system. In their raw states, the datasets consist of low-level student interactions within ASSISTments, with each row of both datasets representing a single action taken by a student; these actions include, for example, attempts to answer a question or requests for system-provided tutoring in the form of hints or scaffolded problems, additional timing (e.g. time since last action) and

Table 5.1: The number of instances and distribution of labels across each outcome.

Label	Number of Instances	Percentage of Positive Class
Off-Task Behavior	568	24.6%
Gaming the System	62	6.0%
Confusion	121	3.9%
Engaged Concentration	2552	82.5%
Boredom	308	10.0%
Frustration	112	3.6%

content-based descriptives (e.g. the skill or knowledge component associated with the problem).

Each of the previous models developed using this data, as cited here and described in the Background Section, utilized a set of features that were engineered from the raw action-level data recorded for each student. In addition to the interaction data collected (e.g. number of attempts, timing, and hint usage), the features also incorporate skill- or knowledge component-level information as well as when the student was working (e.g. during or outside of school hours).

The ground-truth labels of both student affect and off-task behavior were collected using quantitative field observations following the Baker Rodrigo Ocumpaugh Monitoring Protocol (BROMP) [BOAss]. Using this method, human coders observe individual students interacting with the learning software over a short time period (traditionally up to 20 seconds per student) in a round-robin manner. The human coder observes students and applies a label describing the first identifiable affective state as well as the first identifiable behavior from a set including on-task behavior, off-task behavior, and gaming the system as either on- or off-task. Although other projects have observed a range of affective states using BROMP, the labels collected for this work included only four observed states: engaged concentration, boredom,

confusion, and frustration [BDRG10]. The protocol does also allow for uncertainty, where undetermined or observations of affective states or behaviors outside of these sets four are identified and omitted from the data.

We analyze gaming the system using a data set collected via text replays [BCW06]. The ground-truth labels of student gaming were collected using post-hoc examinations of sequences of student log data following a set of previously-developed criteria outlined in [BCW06].

The number of samples corresponding to each label along with the percentage of samples containing the positive class (e.g. the percentage of cases where the affective state or disengaged behavior occurred compared to the total number of labeled samples) is reported in Table 5.1. From this, it can be seen that there are large imbalances among the labels.

5.4 Methodology

As previously described, Jiang et al. [JBB+18] compared two feature engineering methods, expert-generated and deep learning-based, for the development of affect and off-task behavior detectors within the Betty's Brain learning system. Aside from simply testing the generalizability of their findings to more detectors built within a different learning platform, it is the goal of this work to further explore the strengths and weaknesses of each method of generating features. While that previous work utilized deep learning, as is also done in this work, we additionally explore the use of the modeling techniques of co-training [BM98] and multi-task learning [Car97] to observe how these methods may benefit from one type of feature set over another. As such, for each of the detectors of student affect, off-task behavior, and gaming the system, we compare 5 different models utilizing either expert-engineered features,

machine-learned features, or the combination of these feature generation methods, both with and without the use of semi-supervised co-training. The remainder of this section is dedicated to describing each of these 5 methods in greater detail.

5.4.1 Utilizing Expert-Engineered Features

The first set of models use expert-engineered features to detect each label of student affect, off-task behavior, and gaming the system using methods similar to those implemented in previous works. As described in the Data and Labels Section, the expert features are first generated using the the raw action-level log data. In both sets, the features are generated to describe the actions that occur in 20 seconds of observation but also include neighboring actions that go beyond those 20 seconds to capture the full context of these 20 seconds (e.g. a student may take over a minute to respond after receiving help feedback, and we include that response). Therefore, clips are not completely uniform in their duration and can describe intervals longer than 20 seconds, particularly if a student exhibits idle periods while interacting with the system.

In the case of the engineered features used in the affect and off-task behavior detectors, 23 distinct features are created from the raw logs and then an average, sum, min, and max is applied to each action to aggregate these features across each clip (23 distinct features multiplied by the four functions yields the final 92 features). Each set of features describes one or more actions and include such measures as time on task, hint usage, correctness, and other similar descriptives of student performance and interaction with the system, but also include skill-based features (e.g. the number of problems previously seen by the student pertaining to a given knowledge component), and recent performance history (e.g. number of incorrect responses over the last 5 problems).

The engineered features used in the gaming detector similarly aggregate student actions to 20-second clips, but then apply several behavior- and pattern-matching techniques to generate the 33 distinct features. These features attempt to measure gaming behavior through estimates of student timing information (e.g. apparent lack of time spent thinking before asking for help), repetitive actions (e.g. providing the same incorrect response multiple times), and uses the Levenshtein distance [Lev66] applied to the entered text of student responses to identify a specific form of guessing behavior (e.g. providing similar incorrect answers).

Previous work exploring each of these labels applied a large range of rule-, tree-, and regression-based models. For the purpose of the comparisons described in this work, we apply a Naive Bayes classifier, a REP tree classifier (a type of decision tree classifier with reduced error pruning [EK01]), and a Long-Short Term Memory (LSTM) deep learning network [HS97] for the gaming, off-task behavior, and affect detection tasks respectively in accordance with previous works. These models, to the authors' knowledge, represent the highest performing previously published models of their respective outcome measure and were for this reason chosen for comparison; the use of a deep learning model for affect inherently conflates the use of expert-features (used as input to the model) and machine-learned features (through the hidden layer of the network), but we still compare this alongside the other models utilizing expert-engineered features as it is this set of features that is used as input to the model.

As was the case in previous work, each model uses only the clips with corresponding labels as input and produces a continuous-valued output representing the probability that each affective state or unproductive behavior is exhibited within the supplied clip. In the case of off-task behavior and gaming models, each clip is supplied to the respective REP tree and Naive Bayes model and the result is compared

to the binomial label, with positive labels corresponding to each case of off-task behavior and gaming and negative labels corresponding with a lack of each behavior (e.g. on-task behavior and non-gaming behavior). Due to the large number of features generated and likely co-linear relationship between some of the engineered features, a forward feature selection is applied directly prior to each model training procedure to select at most the best 10 features to use in each model.

This paradigm differs for the case of the affect detector model as each of the four affective states are modeled simultaneously as a multinomial classification task through the use of the LSTM model. As a type of recurrent neural network, LSTMs attempt to model sequential relationships within the data; the labelled clips are therefore not treated as independent samples by the model, but rather as a sequence for which a sequence of 4-valued predictions are generated in a many-to-many (or sequence-to-sequence as it is more commonly referred) manner. As was performed in [BBH17] to ensure better temporal consistency within each sequence of clips, student sequences are partitioned such that subsequent clips in the observed sequence occur no more than approximately 5 minutes from the previous clip; spans between clips greater than this threshold are split into two (or more) sequences of student interaction for input to the model.

Each of the models are trained and evaluated using stratified 10-fold student-level cross validation. Given that there is a large imbalance among each of the labels, we stratified each fold based on the number of occurrences of positive labels of each outcome label at the student level in order to generate the folds of the cross validation. This helps to ensure that each fold contains a representative distribution of labels; as this is performed at the student level, it is difficult to produce perfectly balanced folds such that each contains a fully representative set of labels, but the stratification method is an effort toward this property. All subsequent mod-

els described in this work utilized the same student folds described here for better comparability between methods. Each method is trained and evaluated on the same student data and labels within each respective fold.

5.4.2 Deep Learning Models

Unlike the expert-engineered features, the machine learned feature set uses the raw action logs of each student, ignoring the clips and clip-level features described in the previous section. For this feature set, a LSTM model is applied over the raw data to predict each outcome using a set of uninterpretable features learned within the hidden layer of the network. One potential drawback of using a LSTM model in this way is that it assumes that each timestep in the given sequence (i.e. each action taken by each student) occurs at regular intervals which, of course, is not the case. Therefore, to reduce the variance of this interval, a similar practice as was applied to the affect detector model using expert-engineered features. This allows the model to divide sequences of student actions where long intervals may occur between subsequent actions; where the amount of time between two subsequent actions of the same student is greater than 5 minutes, the sequence is divided into two smaller sequences to be input into the model.

Each model utilized the same raw action-level log data that was used to generate the expert-engineered features described in the previous section. In addition to the interaction descriptors such as response correctness and whether the student requested a hint, the knowledge component associated with each problem was also included as a large 1-hot encoded vector in an effort to supply these LSTM models with the same information with which the expert-generated features had access. In addition to these described action logs, the set of features supplied to the gaming model included an additional field corresponding to the computed Levenshtein

distance of each students sequence of incorrect responses (where such sequences of incorrect responses existed) within each problem as was computed for the expert-engineered features of this detector. We incorporated this feature to provide consistency in the information that is exposed to both the machine learning model and the expert feature-engineering process, although we acknowledge that the feature itself is a transformation of the raw responses (i.e. can be described as an expert feature) and was only found to be predictive of gaming the system through prior work exploring the development of expert features for this task [PdCBO14].

Each of the three LSTM models created for each label of student affect, off-task behavior, and gaming the system followed the same general structure comprised of an input layer feeding into a fully-connected recurrent hidden layer of 200 LSTM nodes, and then feeding into an output layer of either 2 nodes (corresponding to a 1-hot encoded positive and negative indicator of either off-task behavior or gaming the system) or 4 nodes (corresponding to a 1-hot encoded vector with one value per observed affective state). The purpose of the hidden layer is to learn a set of 200 features from the raw action logs that are predictive of each outcome label. The commonly applied technique of dropout [SHK+14] is applied between the hidden and output layers of the network in an attempt to reduce overfitting. In all cases, a softmax activation function is applied to the output of each model and trained using multiclass cross entropy [DC97].

The models produce an estimate of each affective state and behavior at each timestep in a sequence-to-sequence manner. In other words, an estimate of each outcome is produced for each action taken by the student. As the labels of each outcome were provided at the 20 second clip-level, the labels are applied to the last action that would have existed in each clip. This allows for a fair comparison between the models utilizing these different feature sets despite each observing data

Table 5.2: Comparison of feature sets across each of the detector models.

Outcome	**Feature Set**	**Model**	**AUC**	**Kappa**
Off-Task	Expert	REP Tree	.734	.352
	Machine Learned	LSTM	.657	.073
	Machine Learned Expert	REP Tree	.753	.400
Gaming	Expert	Naive Bayes	.774	.362
	Machine Learned	LSTM	.542	-.005
	Machine Learned Expert	Naive Bayes	.774	.290
Affect (Collectively)	Expert	LSTM	.760	.172
	Machine Learned	LSTM	.695	.041
	Machine Learned Expert	LSTM	.662	.043
Confusion	Expert	LSTM	.730	.042
	Machine Learned	LSTM	.666	.042
	Machine Learned Expert	LSTM	.609	.01
Engaged Concentration	Expert	LSTM	.775	.281
	Machine Learned	LSTM	.713	.210
	Machine Learned Expert	LSTM	.671	.188
Boredom	Expert	LSTM	.775	.148
	Machine Learned	LSTM	.690	.137
	Machine Learned Expert	LSTM	.677	.041
Frustration	Expert	LSTM	.761	.054
	Machine Learned	LSTM	.713	.060
	Machine Learned Expert	LSTM	.689	.019

at a different granularity; the models are evaluated using the same outcome labels supplied at the same relative points in each student's interaction logs. The models are evaluated using the same 10 folds and cross validation approach as was used by the models utilizing the expert-engineered features.

5.4.3 Machine-Learned Expert-Inspired Features

The third feature set proposed for comparison combines aspects of both expert-engineered features and machine learning. Expert features may be able to help guide a machine learning model to learn better sets of features than either method individually. In addition, since each set of expert features were presumably developed with a particular set of outcome measures in mind (e.g. the features used in the gaming detector were engineered to match the operators used by an expert coder, based on extensive interviews and process modeling in partnership with that coder – cf. [PdCBO14]), such labels may also be able to help guide a machine learning model to produce meaningful, albeit uninterpretable, sets of features to detect such behaviors and affective states.

Specifically, this method utilizes a 2-step training process for a machine learning model. First, an LSTM model is built to use the raw action-level logs as input (just as was done in the previous section for the models utilizing machine learned features), but in addition to predicting each label, the model is trained to predict the set of expert-engineered features as a multi-task learning problem [Car97]. As the affect and off-task behavior detectors utilize the same set of action logs and expert-engineered features, we build one model to read the interaction logs. This model will predict each of the set of expert-engineered features corresponding with the given set of actions, the affective state label, and the off-task behavior label simultaneously. Similarly, for the gaming detector, the raw actions are supplied as

input to a LSTM model that predicts both the set of expert-engineered features and the gaming labels. In this way, the hidden layer of the respective models is regularized to learn a set of features that is both able to construct the set of expert features (although likely with some error) as well as predict the outcome labels for which the features are intended.

Once these two LSTM models are trained - one for the affect and off-task behavior detectors and one for the gaming detector - the hidden layer is extracted and used as the third and final set of features compared in this work. This feature set, referred to as "machine-learned expert-inspired" features, is then supplied as input to each of the respective models used in previous work (i.e. it is used as input to each a Naive Bayes, REP tree, and LSTM as models for gaming, off-task behavior, and affect respectively).

5.4.4 Exploring the use of Co-Training

As described earlier in this paper, it is difficult to fully explore and compare methods of generating features without also considering aspects of the modeling process. This could, of course, refer to the selection of the models themselves, but also is intended to refer to other modeling techniques that may highlight potential strengths and weaknesses of feature sets. We hypothesize that co-training is one such modeling technique.

Co-training is a semi-supervised modeling method that incorporates both unlabeled and labeled instances during the model training process. Given the nature of the observation-based label collection procedure, not all examples in our data (whether considering actions or clips) has an associated affect or behavior label. While there are several modeling methods that exist to incorporate this unlabeled data into each model, the already-described LSTM model inherently allows for this

Table 5.3: Comparison of feature sets across each of the detector models using co-training. All detectors in this analysis uses an LSTM model. *The machine learned model of each detector utilized co-training across actions and therefore mirrors the respective rows in Table 5.2.

Outcome	**Feature Set**	**AUC**	**Kappa**
Off-Task	Expert	.796	.369
	Machine Learned*	.657	.073
	Machine Learned Expert	.781	.405
Gaming	Expert	.856	.180
	Machine Learned*	.542	-.005
	Machine Learned Expert	.847	.327
Affect (Collectively)	Expert	.777	.112
	Machine Learned*	.695	.041
	Machine Learned Expert	.607	.037
Confusion	Expert	.762	.059
	Machine Learned*	.666	.042
	Machine Learned Expert	.596	.018
Engaged Concentration	Expert	.791	.289
	Machine Learned*	.713	.210
	Machine Learned Expert	.611	.090
Boredom	Expert	.783	.178
	Machine Learned*	.690	.137
	Machine Learned Expert	.613	.005
Frustration	Expert	.772	.050
	Machine Learned*	.713	.060
	Machine Learned Expert	.609	.026

co-training to occur given its sequential structure. The model uses the current supplied timestep along with a learned-aggregation of previous time steps in order to better inform each prediction. In fact, this co-training procedure was already used for the LSTM models using the machine learned features; as described, a label does not exist for each action, yet the LSTM model uses information from all previous time steps to predict the respective outcome label where one exists in the given student sequence.

We therefore utilize each of the described feature sets in a separate set of LSTM models that observes the sequence of labeled and unlabeled clips (or actions in the

case of the already-described machine learning feature models).

5.5 Results and Discussion

We compare the results of each set of models within each of the three outcome measures of affect, off-task behavior, and gaming behavior using the metrics of AUC ROC and Cohen's Kappa; in the case of affect, AUC ROC is calculated using a multi-class variant of the metric [HT01], while Kappa is calculated as multi-class Cohen's Kappa, while the models of off-task behavior and gaming use an optimized form of Kappa by learning an optimal decision (0,1) threshold using the training set of each respective fold within the cross validation. Higher values of either metric are indicative of higher model performance with AUC values at 0.5 and Kappa values at 0 indicating chance performance.

The results of each model is reported in Table 5.2, partitioned by outcome measure. From those results, it becomes apparent that, compared to the models utilizing machine learned features, the expert-engineered features lead to notably higher performance across all the outcome labels in regard to both AUC and Kappa. When comparing the performance of the models using the machine-learned expert features (our proposed third feature set), the difference in performance is not as dramatic, but does still lean in favor of the expert-engineered features leading to superior models. By contrast, the machine learned expert features did lead to models that outperform those using expert-engineered features in regard to off-task behavior in terms of both AUC and kappa and is equal in regard to detecting gaming in terms of AUC, but appears to perform less well in detecting student affect.

When comparing the co-trained models using each of these three feature sets, reported in Table 5.3, a similar trend emerges. The use of expert-engineered features

to construct the co-trained models leads to consistently higher performance in comparison to the models utilizing machine-learned features. The co-training models using expert-engineered features also performed better than our proposed machine learned expert-inspired features for all labels in terms of AUC, but obtained lower kappa for off-task behavior and gaming the system.

Despite the small number of cases where the models trained from expert-engineered features did not outperform the others in either AUC or Kappa, these models exhibited consistently high performance in both metrics across all outcomes; while not particularly high, the models performed comparatively well on even the affective states of frustration and confusion, where all models exhibited low values of Kappa. It is for this reason that we can conclude that the use of expert-engineered features led to superior detectors of off-task behavior, gaming the system, and student affective state when compared to using a machine learning approach.

In comparing the results across both Tables 5.2 and 5.3, it can further be concluded that, particularly for the models utilizing expert-engineered features, co-training led to higher AUC than the non-co-trained variant of each detector. This, however, was not always the case in regard to Kappa, where the co-trained models of gaming and affect using expert-engineered features exhibited notably lower values despite the improvement in AUC. This disagreement suggests that the co-trained models have a slightly higher difficulty in identifying gaming behavior or the specific affective state despite exhibiting higher ability to distinguish the two categories across thresholds; this disagreement on a binary label such as gaming could also suggest that the optimal rounding threshold used to classify students differs between training and test sets.

The higher performance exhibited by the model combining the use of expert-features and co-training highlights a potential trade-off of performance and inter-

pretability. While this trade-off was introduced earlier in this paper, the lack of interpretability is not due to the set of features used, but rather in the modeling technique applied; the co-training model, being a deep learning LSTM, falls prey to the same problems faced by automatically distilled features derived through a machine learning approach. In this way, it is difficult to gain an understanding of how the expert-features are being transformed by the co-trained model in order to produce each estimate. Therefore, it is difficult to study these co-trained models to understand more about the behaviors themselves, but the estimates themselves may be useful in other research. Just as previous detectors were used to understand the relationship between affect and disengaged behavior and longer-term learning outcomes [PBSP+13], the estimates produced by the co-trained model can be used in the same way. Conversely, the non-co-trained models can still be used to study the specific affective states and behaviors, as their performance was only marginally lower on average compared to the co-trained models.

5.6 Limitations and Future Work

The research in this paper was limited to data collected within the ASSISTments learning platform, but as described in the Background Section, similar detectors have been developed for a range of platforms. Similar comparisons could be made across these platforms to observe how well these results generalize; small differences in how features are engineered or even recorded by the system may lead to different results.

The method of model co-training in this work also exhibits a limitation in that the LSTM model can only observed unlabeled data within a single sequence. This aspect would not allow the model, for example, to utilize unlabeled data from other

students or sources. Future work may be able to leverage clustering methods or other techniques to allow for more generalizable co-training to benefit the models.

Among the detectors, it was also found in some cases that there was disagreement between the metrics used; the co-trained model of affect exhibited higher AUC than the non-co-trained model, but a lower value of Kappa. Previous work has explored this case across several commonly-applied metrics [BP18], but further work is needed to further explore and leverage modeling techniques to produce detectors that exhibit high performance across all these metrics.

The use of the highest-performing models across each of these detectors can also be used in future research to study other aspects of student learning. This extends beyond the already-discussed application in predicting student longer-term outcomes, and includes the study of other aspects such as the dynamics (e.g. transitions between states and behaviors) [DG12] and chronometry (e.g. how long students

remain in a single state or behavior) [DG11], studied using affect detectors in prior work [BBOH18].

5.7 Conclusions

This work investigates whether expert-engineered features lead to higher performing detectors of student affect and disengaged behavior as compared to using automatically-distilled features learned through a machine-learning approach. We found that the use of expert features led to the most consistently high-performing models. Using co-training as well seemed to lead to even better models in most cases.

The use of expert-engineering to develop features, while perhaps more difficult in regard to time, effort, and likely cost, does appear to lead to greater benefits than simply applying a machine learning model to automatically distill features from the raw data, based on the results found in this work.

Part II

Using Detectors of Student Knowledge, Behavior, and Affect to Drive Action

Chapter 6

The ASSISTments TestBed: Opportunities and Challenges of Experimentation in Online Learning Platforms

Botelho, A.F., Sales, A.C., Patikorn, T., & Heffernan, N.T. (2019). The ASSISTments TestBed: Opportunities and Challenges of Experimentation in Online Learning Platforms. In *LAK 2019 Workshop on Learning Analytic Services to Support Personalized Learning and Assessment at Scale, Tempe, AZ.*

Abstract

The ASSISTments TestBed is a platform for conducting small-scale, short term randomized trials within the ASSISTments online learning platform. Any education researcher may propose an experiment, which will be run at no cost. As a learning system, ASSISTments is positioned to augment teacher instruction and help students learn. As a shared scientific instrument, the

system aims to facilitate the running of studies to learn what types of instructional strategies and content helps which students most and openly share such information and tools to benefit educational research. Through the exploration and analysis of 9 experiments run within ASSISTments, we describe how these tools are being combined with multiple methods to better identify what works for whom. Toward the goal of more precisely measuring treatment effects, this paper acts as an overview of some of the scientific and statistical opportunities that the TestBed system affords when compared to traditional randomized trials in education. We will argue that this framework represents a promising, if uncharted, avenue in the science of education, and merits the attention of both methodologists and substantive education researchers.

6.1 Introduction

The benefits and opportunities made possible through computer-based learning platforms such as ASSISTments extend beyond scientific discovery to include much more practical applications by providing the means to learn what content and instructional practices lead to better student learning. The running of randomized controlled trials has long been the quintessential method of determining the causality of an intervention, and is only augmented through such computer-based systems. The benefit of running RCTs within such systems is not limited to just the scale of the population of students that can be included in a conducted trial, although this too can provide sufficient statistical power beyond what traditional orchestrated studies commonly observe, but rather the benefit is truly in the breadth of data collected for each student, consistency of measures as recorded within the platform, and depth of historical data available within the system that can be leveraged to learn what works best for whom.

A focus on developing methods to more precisely estimate treatment effects is essential in identifying instruction that may be more effective for one group of students than another, and a significant amount of research has been devoted to discovering and developing interventions with heterogeneous effects. Other fields such as marketing and economics arguably have an even longer history of this research leading to methods aimed at measuring such effects [WA18]. Paying attention to context can help identify the situations and for which subgroups a treatment may have an effect to incorporate more personalized interventions to help students.

Through a series of descriptive and empirical examples using 9 studies run within ASSISTments, the goal of this work is to highlight the importance of developing infrastructure to support the running of randomized controlled trials for the purpose of discovering which instructional practices work, and highlight several methods being applied to more precisely measure treatment effects toward the goal of identifying heterogeneous effects where they may exist.

6.2 The ASSISTments Ecosystem

The use of computer-based learning platforms in real classroom settings offer the opportunity to not only test and learn what content and instructional practices benefit students, but also to complete the loop by then deploying successful interventions back to students. It is in this iterative feedback loop that these systems are, at least in theory, able to grow and eventually be able to adapt to meet the needs of students.

The primary goal of this paper is to describe the types of benefits a computer-based learning platform can offer in facilitating scientific discovery and turning research into practice, using a system called ASSISTments to exemplify these op-

portunities. ASSISTments is a free web-based learning platform made available through Worcester Polytechnic Institute. It is used by teachers and students across the United States for homework and classwork, and has been shown to nearly double student learning over the course of a school year as compared to traditional teaching methods [RFMM16].

The whole of ASSISTments extends beyond a computer-based learning system to form an ecosystem [HH14] of tools that are focused on providing immediate feedback to students in an effort to augment the teachers ability to provide instruction in a more data-driven procedure. This teacher-focused approach allows teachers using the system to follow the same curricula as would otherwise be used, but, as students are working on the content within the system, immediate correctness feedback can be provided in addition to other forms of aid including hints and scaffolding where such content has been authored (this additional aid is also pertinent to the idea of conducting trials to learn what types of content benefits students most and will be addressed further in the next section). Even without additional student aid, however, just immediate correctness feedback can help a student understand where he/she needs additional instruction and, through reports provided to teachers through ASSISTments, the teacher can too understand where students need further support; instead of going over homework during class, the teacher can know what content was most troublesome for students beforehand and direct time, attention, and remedial instruction during class to address these areas.

6.2.1 The ASSISTments Testbed

Aside from these attributes that exemplify how a system such as ASSISTments can be used to run RCTs, it is important to further describe the ASSISTments Testbed as this tool extends these benefits to researchers external to the developers of the

platform. The testbed defines a process and set of tools that allow researchers to propose, build, and run RCTs through ASSISTments, and also open supplies the researchers with the Assessment of Learning Infrastructure (ALI) tool [OSW+16] that provides a series of automated analyses and access to the anonymized data from the system associated with their study. The testbed therefore provides researchers with the tools necessary for each aspect of the study design and deployment processes as well as aids in the analyses of such studies; the tool has facilitated over a dozen studies since its deployment resulting in several notable published studies [Fyf16][KM16][MTL+17].

The ASSISTments Testbed defines a set of 5 steps aimed to guide researchers who wish to propose a study from a research idea through to the publication phase of that study. In this way, its goal is to facilitate the running of randomized controlled trials and openly publishing upon the findings. The aim of the testbed is to make it easy for researchers, both those working with ASSISTments and others external to Worcester Polytechnic Institute from where the system is provided, to run numerous RCTs to test the effectiveness of different learning interventions with teachers and students using the software in real classroom settings. In addition, this further makes it easier to replicate studies on different populations and content within the system, as will be the basis of the example analyses described in the later sections of this paper. The testbed and reporting infrastructure also acts as the facilitator of the 9 studies exemplified in this work to illustrate the benefits and opportunities made possible through computer-based systems. The next section describes these studies in larger detail.

6.3 Video vs. Text Feedback: A Case Study of RCEs within ASSISTments

To give a better idea of the process through which a study can be proposed, deployed, and analyzed through the testbed, we will describe the steps using an example intervention. Lets say that a researcher comes to the ASSISTments testbed and wants to run a study to test the effectiveness of video feedback for students as opposed to a text-based explanation given to students who need additional help to learn the material. In other words, the researcher wants to randomize what happens when a student asks for help, giving either a text-based worked example to explain the correct procedure to solve a problem, or a video containing the same information delivered as a video in a more paced manner; this certainly seems like a reasonable comparison as both methods are commonly used in various systems to supplement teacher instruction. With this idea, the researcher proposes to run an RCT within ASSISTments and is given the choice to use the normal population of teachers and students who already use the system for homework and classwork daily, or the researcher can recruit his/her own set of teachers to run a more orchestrated study; for sake of example, we will consider that the researcher chooses to use the teachers and students who normally use ASSISTments. As such, the researcher creates an ASSISTments account and chooses the subject matter on which to run the experiment, and, again for example, lets say that the researcher chooses logarithms as this is a subject that may be difficult for some students and learning what types of aid helps students learn this topic could be meaningful and impactful.

The researcher then creates a problem set using the set of assignment-building tools within ASSISTments aligned to the experimental design; such tools allow the researcher to define, for example, "if-then-else" style and "randomly choose" style

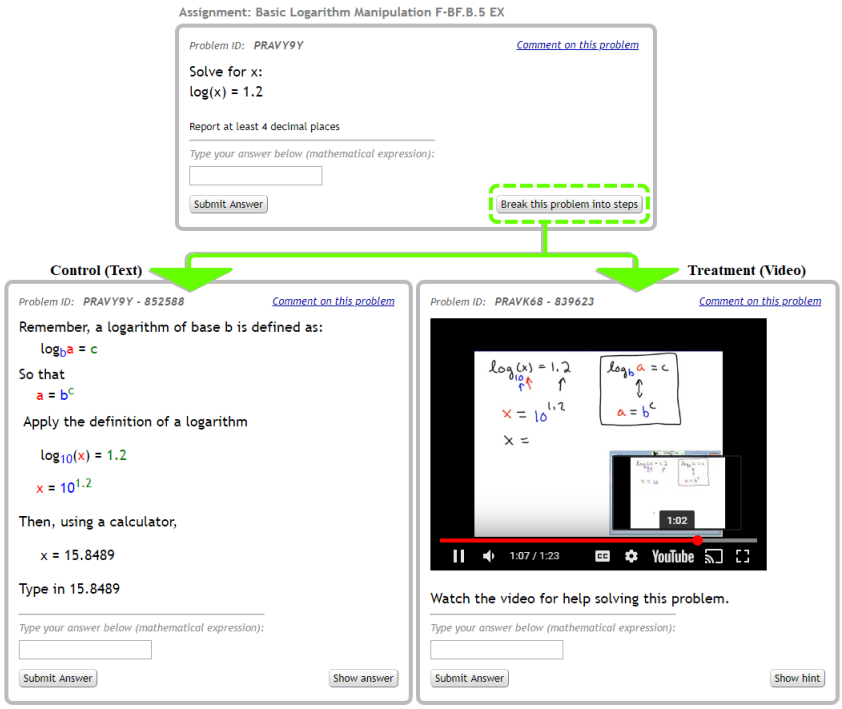

Figure 6.1: An example experimental design with ASSISTments comparing text-based feedback with video-based feedback when students request help.

rules to define where in the problem set randomization occurs. For instance, a reasonable design may first include a question designed to check if students can see video (as some schools may block such content from sites such as YouTube), and only randomized students who have the ability to see video. After this "video-check" the researcher may define a "randomly choose" section that will randomly assign students to either a set of problems containing text feedback or another, almost identical set of problems containing video feedback; an example of such a condition is illustrated in Figure 1, where a student may be randomized to see either a text-based worked example or a video of the same content when requesting help from the system. Of course more complex designs could also include common design elements such as pretests and posttests, but this example will keep the design simple (and it also represents the general design of each of the studies that will be exemplified

in the following section). A problem set created in this way performs student-level randomization, mitigating the need to block students by locale and other factors; although, a researcher may still be able to do so, albeit through a slightly more complex orchestrated design.

Once the problem set is created and approved by a team of researchers and content experts working with ASSISTments (to ensure that the content is not inherently harmful, broken, offensive, or otherwise in violation of IRB terms), the problem set can be deployed amongst the ASSISTments-certified content within the system. While teachers have the ability to create their own content with ASSISTments, many simply choose to use the existing content that has been implemented into the system. When a teacher assigns the particular research-created content, students are randomized and the data is recorded. After a predetermined amount of time, the study is retired and the researcher can begin the planned analyses.

As mentioned above, a tool, called the Assessment of Learning Infrastructure (ALI) aids researchers in the collection and initial analyses of data. Researchers request the data from their experiment by providing ALI with the problem set information and then receive an email containing some initial basic analyses and statistics (e.g. the number of students randomized to each condition as well as completion rates split by condition with a chi-squared test to identify if there is differential attrition between the two conditions). In addition to these descriptives, the researcher gains access to anonymized datasets containing the student data at various granularities including problem-level, action-level, and also student-level covariates generated from data before random assignment to condition (i.e., the students prior percent correct, prior completion, etc.). With this data, the researcher can perform the planned analyses and write the report on their results, citing the initial design document and ALI report to promote open data and science.

This In continuing our example of experimentation through the ASSISTments Testbed, we exemplify a set of nine studies run in ASSISTments comparing text-based and video-based feedback for students. Data from experiments run on the platform of ASSISTments have long been made open and available for researchers to analyze. In 2016, for example, a dataset of 22 such experiments run within the system were published [SPH16] and made open in the hopes that interesting analyses and methods could be applied to better estimate treatment effects and also to motivate other companies and institutes who run RCTs on their own respective platforms to similar see value and make such data open and available. The nine studies observed here are amongst the 22 and are particularly of interest as they apply the same comparison of video versus text feedback. In this way, they act as 9 replications of the same idea and can be used to exemplify some of the challenges and applicable methods available to address such challenges.

These studies were run in mastery-based assignments called "skill builders," where the system provides students with problems until they are able to demonstrate sufficient understanding of the material (e.g., a student must answer three consecutive problems correctly without the use of computer-provided aid), and each student must meet this threshold in order to complete the assignment. Students who are unable to learn the material by the tenth problem are asked to seek additional help, and the assignment is left incomplete (while there are various settings that allow teachers to control each threshold and how to address struggling students, the data used here aligned to the described defaults). We observe the effectiveness of the treatment with regard to the outcome measures of student completion as well as a measure called "inverse mastery speed," calculated as 1 divided by the number of problems needed to complete the skill builder assignment.

6.3.1 Methods to Reduce the Standard Errors of Effects

While ASSISTments and the accompanying ASSISTments Testbed provide infrastructure and tools to run experiments, these alone are not the entire solution to the problem of finding which interventions work for which groups of students. What are missing from these examples thus far are methods that can help to more precisely measure the effects of a particular treatment. Whenever calculating a treatment effect, the ability to accurately measure the impact that the treatment has on any particular outcome is dependent on the magnitude of the effect, but perhaps more importantly, the scale and variance of the population of students included in the study; the more students included in an experiment, the smaller the standard errors on that effect tend to be (i.e. larger samples tend to allow for more precise estimates of the effect). While this goal of reducing standard errors is applicable to any experiment, it becomes much more important to consider when exploring potential heterogeneous effects. If it is difficult to precisely measure a treatment effect across the entire population of students in a particular study, it is much more difficult to measure such effects when observing smaller sub-groups of students.

The next 3 sections therefore describe and compare two methods that are being applied with this specific goal of measuring treatment effects with greater precision. While the examples themselves will not explicitly explore the potential heterogeneity of the interventions, this paper presents some of the pilot work in this area.

Regression to Mediocrity

It is a well-documented issue that a crisis is currently affecting several scientific fields in that, for any number of reasons, experimentation across fields is failing to hold to replication [Ioa05]. If we wanted to know the true effect of video feedback as compared to text feedback on the outcome measure of completion, for example,

due to random variation in content, population, measures, etc., we are likely to observe varying estimates with each replication. In some cases, a replicated effect may appear to have a statistically reliable positive result, while another may show the exact opposite, with many others may show no statistical reliability.

A range of statistics research has been devoted to this and similar problems [Rub81], but the concept for which we are focusing is that of "shrinkage" [EM73]. Also referred to as regression toward mediocrity (or regression to the mean) [Gal86][S+90], the idea is that if we run multiple replications, sometimes our estimate will be too high and other times too low; as we run more replications, the average of our estimates will begin to regress toward the average true effect. Other work has been inspired by the same idea, attempting to use the consistency of data collected across experiments to increase power in estimating effects for individual experiments [PSB+17]. Here, however, we describe a different approach called "partial pooling." The idea of this method is, instead of analyzing each experiment individually and independently, we can pool together similar experiments that we think should have the same effect at once (e.g. replications of the same or similar treatment) in order to better estimate the effects for all pooled experiments. Partial pooling reduces the variance of the estimated effect size of each experiment by looking at the variances and the estimates effect sizes of other experiments, causing the new estimates to shrink toward the mean of the estimated effect distribution.

A drawback to this approach, however, is that it does bias the new estimates toward the overall mean; such is, after all, the purpose as the mean of effect estimates is believed to be a closer estimate to the true effect. Despite this, yet another method may be used to better estimate effects without such a bias. We describe this method in the next section.

A Role for the Remnant: A Model-Based Approach

The idea of applying partial pooling works well in the case of computer-based systems running experiments due to the consistency of measures collected across students (although the method itself is not inherently limited to cases where the measures are as consistent as used here), as the system records the same information for each student. However, this is also true for all students using the system, not just those who participate in an experiment. So what, then, can be learned from all the students who are not randomized to condition? In the case of ASSISTments, there are hundreds if not thousands of students using the system every day, and if we could utilize their data to better analyze experiments, the added power is likely to help reduce standard errors on the estimated treatment effect.

Previous work explored the use of this population of students external to the experiment, which has been referred to as the "remnant," to more accurately estimate treatment effects [SBPH18]. The remnant essentially consists of all students who have ever used the system that were not a part of any of the current experiments under analysis; they may have been a part of previous experiments but, for instance in the case of our example, it includes a large sample of students disjoint from those who participated in any of the 9 example RCTs. But what, if anything, can be learned from this group? No randomization occurred for these students, and there is no guarantee that a condition in the experiment is "normal" behavior, meaning that the manner in which students interacted with the system during the experiment as compared to normal usage may be very different. What we do know, however, is that data pertaining to outcomes of interest (i.e. assignment completion, knowledge level and correctness, number of problems needed to complete mastery-based assignments) is available for the remnant as well as those in the experiment.

It is from this idea that a method called "remnant based residualization," or

REBAR [SHR18], was developed. The process is rather intuitive. First, we can build a model using the remnant to predict an outcome measure of interest. In our example case, we use the remnant to train a model to predict whether a student will complete an arbitrary next assignment. Second, the trained model is applied to predict the outcome measure for those in the experiment. Third, the estimates of the model (our prediction of whether each student will complete the experimental assignment), are subtracted from the actual outcome; this step is essentially removing variance from the outcome measure of interest that can be explained away by the model trained on the remnant. From this point, the last step is to simply analyze the experiment using any desired method using the residual in place of the actual outcome. As the model is trained on a population completely external to the participants in the experiment, the estimates are unbiased. For this reason, the estimates themselves do not even need to be accurate; a bad model should be just as bad for everyone (on average). However, the better the model is at predicting the outcome, the more variance that can be accounted for within the experiment leading to more accurate treatment effect estimates.

Why Not Both?

As mentioned in the previous section, the last step of the REBAR method uses the residual to run any set of desired analyses. For this reason, the REBAR process and the described partial pooling method are disjoint approaches and therefore could be combined to even further reduce standard errors of the estimated treatment effects. In this way, we can take advantage of both the scale and breadth of data made available through the use of the remnant, while also taking advantage of the consistency of measures across the experiments.

We use the model estimates from the REBAR method for both outcomes mea-

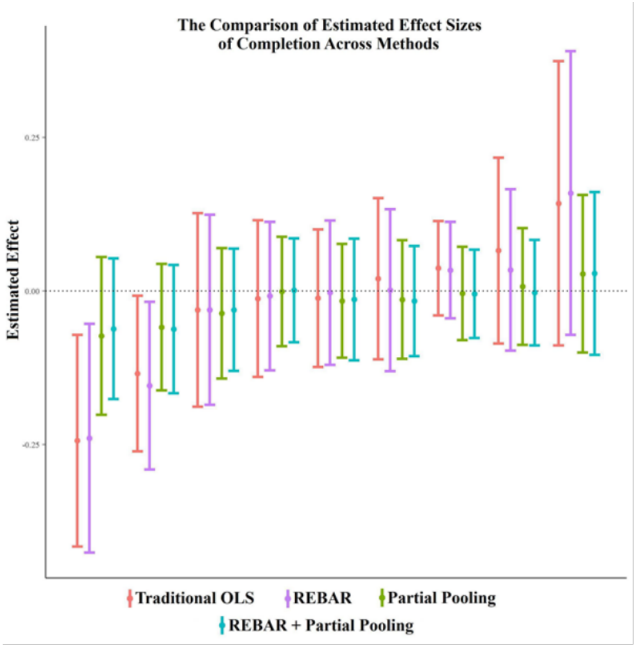

Figure 6.2: The estimated treatment effects of student completion for each of the 9 experiments across all methods.

sures of completion and inverse mastery speed as described in the previous section. The estimates are subtracted from the observed outcomes following the REBAR methodology, and then the resulting residual is used in the Bayesian partial pooling approach. The combination of these two approaches results in the reduction of standard errors across all example studies. As shown in Figure 2, the combination of methods reduces the standard errors of all experiments when compared to the traditional method and is superior or at least comparably similar to either method alone.

In consideration of the second outcome measure of inverse mastery speed, the combination of methods again leads to considerable reductions of standard errors beyond that of the traditional method in all experiments, as seen in Figure 3. Similarly to that of Figure 2, the combined method performs better or comparably similar to either other method alone.

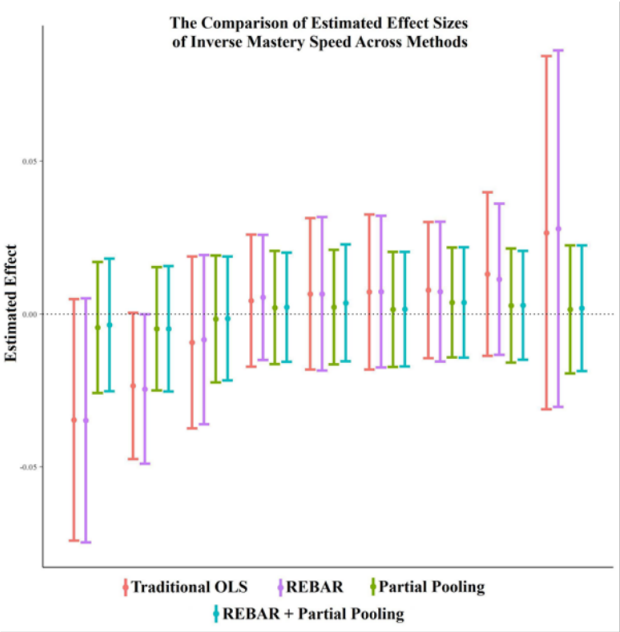

Figure 6.3: The estimated treatment effects of student completion for each of the 9 experiments across all methods.

It can be seen in both analyses, however, that the combined method does not lead to the smallest standard errors in every case. It is important to explore and understand, as is the goal of ongoing and future work, when each method is likely to lead to improvements in precision. Regardless, these methods show promise in their ability to aid in the analysis of experiments and discovery of potential heterogeneity in the measured effects. It is also the case that the methods helped to remove some of the variation of the 9 replicated studies, where the combined method no longer results in statistically reliable effects in any of the experiments; it is important to emphasize, however, that this is largely influenced by the partial pooling methods bias toward the mean effect measured across all experiments and that these particular experiments were chosen for these analyses in-part for exemplary purposes. Ongoing and future work is further exploring the application of these methods at larger scale across multiple experiments running through the ASSISTments Testbed.

6.4 conclusions

The issues and challenges faced by the field as it moves toward new experimental environments and, through these, new data environments, are by no means novel, but rather tools such as computer-based platforms are merely allowing us as researchers to finally address these problems in more practical ways. It has always been a challenge to design replicable RCTs to test ideas; this is a challenge for replicability of results (i.e. the same or similar findings and conclusions are reached after additional trials), but also in a much more direct interpretation of replicability, where a design can be replicable. Computer-based systems offer new ways to allow for clear replication, using the same design for new populations or contexts, using the same measures calculated in the same ways across all experiments. This consistency alone offers new opportunities to more accurately evaluate instructional strategies and the like.

Just as replication has been a challenge, the ability to accurately estimate treatment effects is another long-standing issue. It is important to consider how existing methods can best be combined with the opportunities that computer-based systems offer. Where in the past collecting data from several dozen students served as a challenge to any researcher intending to run a randomized controlled trial, it is now more trivial to collect data from several hundred students, if not more, through such systems allowing us to direct more focus to the other prevalent challenges. Issues such as testing ideas in new contexts or identifying heterogeneity become much more feasible as the scale and replicability of studies becomes easier.

We refer to and describe a number of studies and research in this article that have been facilitated by ASSISTments and the ASSISTments Testbed, but these are small examples compared to what is currently possible with these and similar

tools. These tools in combination with the development and application of methods to more precisely measure treatment effects holds great promise in regard to the goal of discovering what works (and what does not work) for particular groups of students.

Chapter 7

Putting Teachers in the Driver's Seat: Using Machine Learning to Personalize Interactions with Students

The following grant proposal was written alongside PI's Neil Heffernan, Hilary Kreisberg, and Jacob Whitehill. This has been funded as NSF #1822830 ($744,317). Supplementary figures and materials as well as references for this proposal are included as appendices at the end of this dissertation document.

Tools for communication are getting smarter; when someone replies to a text message or email, predictive or 'suggested messages' automatically pop-up as quick options. Users can easily select a suggestion rather than spending time writing their own message, thus allowing technology to facilitate human-to-human interactions. Computer science has developed functionalities like machine learning and natural language processing to enhance user-experiences across domains. In business, the

development of prediction systems that offer smart suggestions for what one might want to say next has saved time and enhanced communication, and oftentimes, those reading the message do not know that it was machine generated. However, such means of enhanced communication do not yet exist in most educational contexts. Automated messages that address the diagnosis, tone, and context of specific student actions are a necessary function for teachers as they respond to the influx of data compiled as student work transitions to online environments.

7.1 The Problem

The typical environment of student learning is evolving; where once students relied on textbooks and paper, many now complete their assignments online (Lu, 2008). School administrators, curriculum developers, and most notably, teachers, are frequently asked to utilize online tools in support of instruction. The integration of computer-based systems into existing instructional practices and curricula has led to a recent rise in the usage of online tools and content (Pawlowski & Bick, 2012). In fact, the New York Times (Singer, 2017) reported that, in the last year alone, schools across the United States have purchased eight million Chromebooks.

In mathematics, online programs have started to supplement instruction through adaptive, student-paced assignments (e.g., Carnegie Learning's Cognitive TutorTM or McGraw Hill's ALEKSTM). The term "student-paced" means that content is assigned to students based on how fast they are learning; when students demonstrate mastery, the system accelerates them to the next topic. This differs from "teacher-paced" systems, in which the teacher assigns online content pertinent to a shared classroom focus. Many online learning systems seek to provide "personalized instruction" to students, applying computer-assigned content to a student-paced

learning progression. Although this sounds appealing, "personalization" removes the teacher from the driver's seat and creates a disconnect between teacher instruction and online instruction. This disconnect is alarming considering research has demonstrated that the role a teacher plays in student learning is a pivotal indicator of academic achievement (RAND Education, 2012). Researchers have found that student-teacher interaction quality informs classroom organization and instructional support, which correlate with numerous measures of students' cognitive, emotional, and social engagement as well as their social preferences of their peers (Hughes, Cavell, & Willson, 2001; Rimm-Kaufman, Baroody, Larsen, Curby, & Abry, 2015). If an online learning system 'personalizes' instruction, the teacher is left behind and students are deprived of human support. We argue in our proposal that teacher-paced online systems offer a more powerful way to incorporate computers into the classroom.

Feedback is also an important part of the learning process, as it attempts to align metacognition with performance and supports growth and success. Actionable instruction allows teachers to guide students with the steps necessary for improvement. However, giving personalized feedback can be overwhelming for middle school teachers who are often responsible for more than 100 students. This task becomes increasingly complicated as teachers assign online learning activities and sift through hundreds of responses for each problem assigned. While it is evident that effective feedback from teachers can improve performance by helping students focus on specific steps that can lead to success (Wiliam, 1999), opportunities for teachers to efficiently provide personalized feedback in online learning environments are limited.

We contend that online learning environments need to put teachers back in the driver's seat by using machine learning to help personalize data-driven teacher-

student dialogues. Currently, the frequency and quality of interactions that occur between teachers and individual students are limited by the time available for a teacher to interact with more than 100 students. Teachers who use online tools that collect a plethora of data now face the problem of reading and interpreting massive amounts of student information. Computer-based systems can provide action-by-action reports of how students reached an answer, the time it took to do so, and a wealth of other performance-based information. Although such insight could help teachers provide better instruction and address individual student needs, it can be impossible to wade through such vast data to find meaning. There is simply not enough time for teachers to make use of such data (Bill & Melinda Gates Foundation, 2015), ultimately leading to weakened and impersonal teacher-student relationships and reduced performance, contrary to the inherent goals of these learning platforms. When considering the positive impacts of increasing teacher involvement in the learning environment, it is clear that online learning systems should assist teachers in interpreting student data and responding to students.

Very few online mathematics environments allow students to enter text-based answers, and those that do currently offer no features to enhance a teacher's grading experience. In the cases of McGraw Hill's ALEKSTM and Carnegie Learning's Cognitive TutorTM, there is no concept of an open-response question; their designs restrict problem types to those that can be quantified or automatically graded.While math curriculum increasingly asks students to answer 'why' questions, online learning systems lack the capacity to handle open-response answers. The widely adopted Open Educational Resource (OER) EngageNY is comprised of a large number of problems and problem-types across multiple grade levels, with the highest proportion of those problems (more than 38%) offered as open-response.

A few existing systems do promote computer-interpreted open-responses. For in-

stance, AUTOTutor (Graesser, Chipman, Haynes, & Olney, 2005) promotes student-computer dialogue in which the computer is intended to be believed as a cognizant partner. However, even in these environments, there is no way for teachers to efficiently review and respond to the student's dialogue. The missing link is a system that offers teachers their traditional task of reviewing, while at the same time taking advantage of technologies in computer-based natural language processing (NLP) (e.g., deep learning and memory networks) and methods of understanding student behavior (e.g., affect detectors) to leverage open-response questions in support of data-informed communication and instruction.

7.2 The Opportunity

The previous section detailed a set of problems that have emerged as the adoption of computer-based systems for classwork and homework has tended to minimize the role of the teacher in learning environments. In this section, we will highlight several opportunities that can be leveraged when exploring and developing solutions to the problems posed.

7.2.1 Results of Prior NSF Support: Heffernan's ASSISTments and other CoPIs

ASSISTments.org was created while conducting research for past NSF awards. Heffernan's NSF CAREER award (CAREER: Learning about Learning award #0448319, $646,075, 2006 2013) is the most relevant grant that helped to create ASSISTments. It's intellectual merit included more than four dozen peer-reviewed publications in machine learning, deep learning, clustering, predict etc. (see the separate section in the references noting these 60+ papers). Other NSF grants have also supported AS-

SISTments, including NSF# 0742503 whose intellectual merit included 22 published randomized controlled trials measuring different ways to improve student learning through feedback (see the separate section in the references noting these 22 papers). The broader impact of these awards has been support for thousands of students via the ASSISTments service provided as a free public service by WPI. Last year alone, students solved more than 12 million problems. Over more than a decade, the WPI team has written tens of thousands of questions and teachers have written an additional 75,000 questions for their students.

Whitehill's prior NSF support is as co-PI of an ongoing NSF Cyberlearning grant ("INT: Collaborative Research: Detecting, Predicting and Remediating Student Affect and Grit Using Computer Vision," #1551594, $749,000, 2016-2019). The intellectual merit of this project has taken the form of 3 papers that are listed in their own section of the references. The broader impact is to use neural networks to help solve real-world educational.

Kreisberg does not have any NSF grants on which to report.

The ASSISTments platform allows teachers to enhance any homework assignment with online feedback for students and reports for teachers (see Figure SD1 for assignment questions and the associated item-report). When students solve a problem in ASSISTments, after entering their answer they are told if they were correct. If they got the problem wrong, they can try again, and in some instances, receive additional tutoring. Such data entry allows for the creation of class and student reports geared toward the teacher with the purpose of informing the next day's instruction.

The item-report is designed to provide teachers with information that is easy to respond to. To explain, here is a short vignette of Ms. Kelly, a 7th-grade math teacher (an actual video of Ms. Kelly reviewing an item-report can be seen at Kelly

et al., 2013). Let us assume Ms. Kelly assigned the problems in Figure SD1 to her students using ASSISTments. The next morning, Ms. Kelly prepares for class by assessing the item-report. She would probably want to talk to her students about question 1, as it was challenging (only 27% of students provided the correct answer) and 66% of students who got it wrong responded with an answer of '0'. She realizes that these students made a common error of subtracting -9 from 9 to get zero rather than solving 9 - (-9) correctly to reach 18. She decides to spend a portion of her class time addressing this misconception. She then reads through some of the open-response answers and notices that many students, including Wei and Wakeeta, thought that Mountain Charter was always better, failing to see how one plan is better for more people while one is better for fewer. She also decides to spend a portion of her class time addressing this misconception. However, reading each student's open-response answer could be very time consuming because she has over 100 students.

Although the item-report is an exceptionally helpful tool for teachers, as shown by Ms. Kelly's vignette, the data provided in the item-report is just the tip of the iceberg and yet, can already reach overwhelming levels when interpreted manually. In our solution section, we will describe how we intend to take advantage of all student data in order to support teachers' online dialogues with students.

SRI study that showed ASSISTments caused better learning and closed achievement gaps

Recently a year-long randomized controlled trial conducted by SRI International concluded. This study (Roschelle, Feng, Murphy, & Mason, 2016) produced three main findings: 1) Teachers reliably changed the way they reviewed homework, consistent with the intended-use model. They still spent approximately the same

amount of time reviewing homework, but their reviews were focused on a smaller number of difficult problems rather than all questions. 2) Students in schools randomly assigned to use ASSISTments had reliably higher rates of learning (p¡.008) as shown by an additional eight-point gain on an end-of-year standardized test, an approximate 75% gain atop the 11 points students are expected to gain (on average) in a school year. 3) The intervention helped to close achievement gaps rather than widen them, as Steenbergen-Hu and Cooper (2013) found for most other K-12 mathematics intelligent tutoring systems. Students with incoming 6th-grade scores below the median experienced greater gains than those above the median.

Open Educational Resources (OER) are incorporated into ASSISTments: We have excited districts

Several OER, such as the commonly adopted EngageNY curriculum, have emerged as free and accessible resources for both teachers and students. These 'digital textbooks' are used by districts across the nation, many of which are also using ASSISTments as an instructional support to provide formative assessment opportunities, technology integration, and problem-solving interactions. This overlap is due to the fact that the team at ASSISTments incorporated problems and answers for every homework problem, classwork problem, and exit card available in EngageNY into ASSISTments' certified content. Since ASSISTments already supports OER on its platform, enhancing the use of data collected with additional functionalities will make the product more useful to teachers as classrooms digitize. As you can see from the attached support letters and details in the collaboration and management plan, we have identified a consortium of districts that currently use OER and could adopt our tool in supplementing their approaches to learning. Dr. Kreisberg, who already works closely with these districts, will be managing these districts' partici-

pation to support their desire to continue to use Google Classroom, technology, and OER.

7.2.2 An Opportunity to Apply Google's Smart Reply Technique to the Education Problem

Google's Smart Reply system (Kannan et al., 2016) employs a number of exciting techniques, combining a recurrent deep learning Long Short Term Memory (LSTM) network (Hochreiter & Schmidhuber, 1997), clustering, and semi-supervised learning methodology to generate suggested email responses for users. In short, Google researchers use the LSTM model to generate 10-30 of the most likely responses to a given email, after which the response candidates are compared to generated clusters of messages with shared semantic intent (i.e., messages with the same general meaning despite possible lexical differences) in order to suggest three messages exhibiting sufficient variation.

The heart of the system is a simple LSTM model for response selection using a sequence-to-sequence technique (Sutskever, Vinyals, & Le, 2014). Consistent with the approach of the Neural Conversation Model (Vinyals & Le, 2015), the incoming email message is input word-by-word into a model which then has the ability to output, word-by-word, every possible response to the initial message; as the total number of possible responses is impractically large, the actual number of output messages are limited to the top most likely responses given the input.

Generated clusters each represent a different semantic intent, or general subject matter, that is seeded by a small selection of example cases. For instance, one cluster could have the intent of "thank you," with a few instances being thanks, thanks!, and thank you. Google researchers hand-code examples, three to five instances for each cluster, and use them to compare the LSTM-generated messages to estimate the

semantic intent of unlabeled generated responses. In this way, the process becomes a semi-supervised learning task, where a few labeled responses are used to propagate semantic meaning to unlabeled instances to ensure a breadth of meaning across the final selected messages.

This sequence-to-sequence modeling as an NLP technique is not based on traditional parsing methods. Instead, it is, in a simple sense, similar to trigram poetry, where the selected word is based solely on the two previous words. The sequence-to-sequence model, however, can use a context that spans the entire length of the word sequence and is more sophisticated in that the overall technique does not require traditional natural language parsing methods.

Triggering - the initial step to the methodology employed by Google's Smart Reply - arguably contributes most to the practicality of the system by allowing only three email message options from which the user, if provided with Smart Replies, can choose; this prevents users from learning to ignore Smart Replies. This process attempts to filter out emails that do not warrant a response (negating the need to suggest replies), as well as those that are more sensitive or open-ended to effectively produce viable responses. The method uses a simple feed-forward neural network to produce a value that combines both the probability that a response is necessary and also the confidence of the system to be able to produce responses that would be useful to the user. At the time of publication (Sutskever, Vinyals, & Le, 2014), a reported 11% of emails passed the triggering process and displayed suggested Smart Reply messages. Among applicable emails, 10% of users responded with a Smart Reply, with 40% of these users selecting the reply on the far left, deemed by Google's algorithm to be the most useful. Therefore, one percent of email messages from a Google client utilize Smart Reply - an impressive amount given the scale of Google.

Considering the Smart Reply methodology, we see great opportunity to appro-

priate the promotion of data-informed communication to an educational context. We see a parallel in the problem addressed by the Smart Reply system to the problems identified in Section A of our proposal. The Smart Reply system needs to read and interpret, if only internally, the content within a received email in order to provide a human user with response choices. In classrooms, teachers often act in a similar manner, reading through student homework responses in order to decide what feedback to provide students in order to improve their learning experience. This parallel has inspired our current work.

The Solution section of this proposal will detail our approach to this problem, expanding upon and adapting the methodologies employed by Smart Reply to develop a system that specifically addresses the needs of teachers. The collaboration and management section details Heffernan's and Whitehill's extensive expertise in deep learning, recurrent memory networks, clustering, and other techniques that will allow for the development of such a system.

7.2.3 ASSISTments currently has detectors that rely on student input

As students and teachers began to take advantage of ASSISTments and its feedback, it became evident that more could be accomplished with the high volume of data collected by the system. Through internal (at WPI) and collaborative efforts with several other learner-analytics researchers and institutions, we have a history of peer-reviewed publications focused on the creation, study, and usage of automated, sensor-free detectors of student performance, behavior, and affective state within the ASSISTments learning platform, as detailed in the subsections below.

Category	Detector
Existing Detectors	
Performance	Correctness
	Within-Assignment Learning
	Completion
Affect	Engaged Concentration
	Confusion
	Frustration
	Boredom
Behavior	Wheel Spinning
	Gaming the System
	Mental Effort
	Persistence
	Carelessness
Detectors to be developed	
Student History	Improvement Over Time
Student Open Response	Open Response Understanding
	Open Response Effort

Table 7.1: Categories of detectors to be utilized by DRIVER-SEAT.

Detectors of Student Performance

Several methods pertaining to understanding student knowledge and performance have emerged and subsequently have been studied within the context of ASSISTments.

Deep learning methods, describing a family of techniques utilizing multi-layered neural networks, have exhibited a recent increase in usage in a wide-range of fields due to increased development support, advances in technology, and subsequently promising performance when compared to more traditional methods. A type of deep learning model, known as a recurrent neural network (Williams & Zipser, 1989), has been the basis of several recent works that suggest notable improvements to estimating short-term student performance. The development of the Deep Knowledge Tracing (DKT) model (Peich et al., 2015) was among the first applications of this type of deep learning model within an educational context, reporting

vast improvements over the widely applied models of Bayesian Knowledge Tracing (BKT) (Anderson, Corbett, Koedinger, & Pelletier, 1995) and Performance Factors Analysis (PFA) (Pavlik Jr., Cen, & Koedinger, 2009). Heffernan's team and others (Khajah et al., 2016; Wilson et al., 2016; Xiong, Zhao, VanInwegen, & Beck, 2016) later used the same methods to correct Peich et al.'s overestimation of effects. Unsurprisingly, student correctness has been commonly studied as a measure of student knowledge, as it is among the most basic metrics of student performance for teachers to address, and such models have been built to recognize short-term learning progressions over the length of assignments.

In addition to estimates of student knowledge, completion and persistence have also been studied by implementing these models within ASSISTments data. The different types of assignments available within ASSISTments has allowed researchers to utilize mastery-based assignment data to study productive perseverance (Kai, Almeda, Baker, Heffernan, & Heffernan, in press) and persistence-related measures such as student "wheel spinning." The concept of wheel spinning (Beck & Gong, 2013) is derived from the analogy of a vehicle made immobile due to snow or mud; effort may be applied, where the vehicle spins its wheels, but no progress is achieved. In the context of education, a student may "spin his or her wheels" and apply effort to learn a concept, but make little or no progress toward effectively mastering the material. It is thought that identifying wheel spinning early on can help inform remedial instruction or prevent unnecessary frustration.

Detectors of Student Affect

Students' emotion and affective state have been proven as significant predictors of short- and long-term performance (Craig et al., 2004; Pardos et al., 2014). Using student affect detectors researchers have reliably predicted affect from ASSISTments

logs (San Pedro, Baker, Gowda, & Heffernan, 2013), and have used affect states to better predict state test scores (Pardos, Baker, San Pedro, Gowda, & Gowda, 2014), college attendance (San Pedro, Baker, Bowers, & Heffernan, 2013), STEM-related college majors (San Pedro, Ocumpaugh, Baker, & Heffernan, 2014), and how these detectors generalize across rural, urban, and suburban contexts (Ocumpaugh, Baker, Gowda, Heffernan, & Heffernan, 2014). With such works pointing to the importance of detecting and measuring student affect, the argument for their inclusion in this proposal is well-founded in this prior research.

A significant amount of research has been conducted on the detection of student affect state by aligning ASSISTments data to collected quantitative field observations using the Baker Rodrigo Ocumpaugh Monitoring Protocol (BROMP) (Ocumpaugh, Baker, & Rodrigo, 2015). This protocol allows human coders to observe students in the classroom while working within the learning system and label them based on one of four commonly studied affective states: engaged concentration (Csikszentmihalyi, 1990), frustration (Kort, Reilly, & Picard, 2001; Patrick, Skinner, & Connell, 1993), boredom (Csikszentmihalyi, 1990; Miserandino, 1996), and confusion (Craig, Graesser, Sullins, & Gholson, 2004; Kort, Reilly, & Picard, 2001).

Initial development of sensor-free affect detectors, utilizing only the recorded student log data aligned with human-labeled observations, explored a number of tree-based, rule-based, and Bayesian models, ultimately reporting moderate model accuracy above chance (Ocumpaugh et al., 2014). Later, Wang, Heffernan, and Heffernan (2015), improved upon these initial affect models by incorporating more information pertaining to skill, or knowledge component, as well as class-level features. Most recently, Heffernan and colleagues (Botelho, Baker, & Heffernan, 2017) explored the application of deep learning models, exhibiting a significant increase to model performance. That work compared three variants of recurrent neural

networks - traditional recurrent, LSTM, and Gated Recurrent Unit networks - as sequence-to-sequence models to estimate labeled student affect states.

Detectors of Student Behavior

One of the most informative forms of data that can be provided to teachers is not the end result or performance metric alone, but data that can describe the process that contributed to a result. As such, detectors of student behavior have emerged in the field of learner analytics and, among other systems, their application and further development have been studied using ASSISTments data. One of the more negative behaviors that has been studied is that of students "gaming of the system," or cheating the system, referred to hereafter simply as "gaming." Student gaming is exhibited in a number of ways depending on the type of assignment and availability of computer-provided tutoring. Essentially this behavior is exemplified by a student progressing through an assignment by exploiting an aspect of the system rather than administering effort to learn the material. In such cases, the student may proceed quickly through the assignment, exhausting all computer-provided tutoring to reveal the correct answers (it is common to see these students finishing such assignments in just a few minutes' time, while the rest of the class takes significantly longer depending on the difficulty and number of questions). Developments toward detecting this behavior can help inform teachers that a student has not applied effort and likely does not know the assigned material despite having "completed" the assignment.

More recent work (Botelho, Ostrow, & Heffernan, in submission) has explored student persistence and mental effort as distinct student constructs to explain conflicting ideas of students persisting yet gaming while others apply effort but fail to complete their work. It attempts to address the 'productive' aspect of produc-

tive perseverance, linking mental effort and ultimate completion to define a state of persistence; the interaction of these constructs suggests a spectrum of different student behaviors. While only pilot work has been completed thus far, results suggest promise in regard to developing detectors of mental effort that, alongside existing detectors of gaming and knowledge, may help better describe the student-learning process exhibited while working through assigned material in ASSISTments.

While not yet directly studied within the context of ASSISTments, another developed detector of student behavior observes the construct of carelessness. This construct can help teachers understand the level of attentiveness exhibited by students, as it estimates, in a simple sense, when students answer a problem incorrectly though they actually have sufficient understanding of the material. In this scenario, the students may make a simple mistake that they may have caught if they were more careful and checked their work before submitting their answer.

7.3 The Solution

In this project, we will create the Dialogue Reinforcement Infrastructure for Volitional Exploratory Research - Soliciting Effective Actions from Teachers (DRIVER-SEAT). DRIVER-SEAT is a tool that will enhance personal one-on-one communication between teachers and students. DRIVER-SEAT will allow teachers to use automatically generated, suggested responses to provide personalized feedback to students in the form of a dialogue. We will answer a set of research questions about how to machine-learn dialogue-based systems for teachers (see Section C.4 for details on our research questions). To accomplish this, we will work with a team of twenty pilot teachers and of those, choose five to become development teachers. This work will create a tool that allows teachers to take advantage of the data collected by

technology in order to connect with their students in ways that improve learning. In this section, we will begin by describing DRIVER-SEAT with a vignette, then outline the project activities, describe DRIVER-SEAT in greater detail, and finally, outline the research questions we plan to address.

DRIVER-SEAT needs a cooperating platform from which to pull data and also to send messages to students, and for this we have chosen to develop the tool within ASSISTments. In Year 3, however, we will show that DRIVER-SEAT can be generalized to other systems by testing it within EdX, a platform for hosting free Massive Open Online Courses (MOOCs). We want DRIVER-SEAT to be able to work with, and learn across, any student-level clickstream data. Ideally, clickstream data in the form of IMS Calipers standard (IMS, 2017) or the xAPI (xAPI, 2017) standard should be consumable to help teachers send messages to their students.

7.3.1 DRIVER-SEAT Vignette: What we want this to look like

This vignette will demonstrate a hypothetical use case of DRIVER-SEAT. Let us reconsider Ms. Kelly, who we discussed in Section B.2. Like before, she assigns a set of problems that students work on at home, and she assesses the item-report before class the next day (Figure SD1). She first decides to talk to the whole class about problem 1 and its common wrong answer. When finished, before reading all the students' open-responses, she opens DRIVER-SEAT and looks at the dialogue-initiation interface, as shown in Figure SD2. The interface shows eight students who have been selected as good candidates for a teacher-student dialogue. Each student has been given a diagnosis (as determined by the detectors) that should be addressed by Ms. Kelly. She can select "show" to see the reasons for each diagnosis. In this interface, column 1 provides the name of the student (we have added descriptors to

the names to help the reader understand the diagnosis provided by DRIVER-SEAT), and column 2 provides the diagnosis and links to a popup that shows the students' clickstream data to justify the diagnosis. Column 3 allows Ms. Kelly to decide if she wants to send a message or override the system (if she chooses to override, the system asks for an explanation so the detectors can learn from her response). Columns 4 through six show three automatically suggested messages, similar to Google's Smart Reply. The first is the default, but in some cases Ms. Kelly selects another message or opts writes her own (as shown in column 8, row 2 for Learning Lalit). Column 7 offers the suggested action for the student - DRIVER-SEAT is not just about sending a message, but about starting a dialogue. Students are expected to read the message and complete the assigned action. The next day, Ms. Kelly will assess the follow-up interface that allows her to check on her students' completion of these actions, and ultimately end the dialogue.

Ms. Kelly sees that DRIVER-SEAT has diagnosed Gaming Gangi as "gaming the system." The data displayed when she clicks on "show" outlines that he has not even spent enough time to read each assigned question. DRIVER-SEAT suggests the message, "You seem to be going too fast." Ms. Kelly adopts that default to send Gangi.

For Learning Lalit, DRIVER-SEAT suggests that Lalit should receive positive feedback considering she has shown improvement. DRIVER-SEAT knows that Lalit has had trouble in the past and wants to draw this improvement to Ms. Kelly's attention; it wants teachers to "catch students while they are doing well." Without DRIVER-SEAT, Ms. Kelly may have otherwise overlooked this subtle improvement. Ms. Kelly likes the second phrasing and selects it (from these types of selections, DRIVER-SEAT will attempt to learn the most desired and effective phrasing of messages). Ms. Kelly chooses not to assign an action.

DRIVER-SEAT identifies two students who seemed confused (according to Baker's detectors) and suggests assigning a Skill Builder, or a mastery-based learning assignment, on the missing standard. Notice (on the item-report found in Figure SD1) that Carl got every problem wrong, just like Gaming Gangi, but he took his time. Courtney's diagnosis was also confused. Ms. Kelly was fine with default suggestions for Courtney, but since she sees under "show" that Carl has been diagnosed with confusion five times recently, she decides to deliver her message face-to-face. She selects the "Will talk to him in class" button so the system can learn that the teacher agreed with the diagnosis. She was very glad she had this record of Carl's work, and will be able to show this information to Carl's parents, if necessary.

DRIVER-SEAT also brings Super Sachi to Ms. Kelly's attention as he performed better than usual. She is pleased to catch a student while he is doing well. Ms. Kelly decides to select the first message and send Sachi a message asking him to reflect on his performance and communicate what he did differently this time, to try and instill positive behavioral and cognitive principles.

DRIVER-SEAT identifies two students (Wei and Wakeeta) who did great on the gradable part (questions 1, 2a, 2b, 2c, 2d), but poorly on the open-response (question 2e) by making the same error. They both thought that Mountain Charter was better, failing to even recognize there was a break-even point between the two companies. After selecting the first message, Ms. Kelly gave them the action of evaluating other responses and hopefully selecting one that was correct to show a better understanding of the problem. She will check on their work tomorrow with the follow-up interface.

Unlike Wei and Wakeeta, Linda's response was incorrectly tagged; it looked incorrect because she had responded, "Mountain Charter was better." However, when Ms. Kelly took another look, she noticed that, in fact, Linda had only made the

mistake of not stating the break-even point of 100. She decides not to send the message by selecting the, "I disagree with the diagnosis" checkbox (so DRIVER-SEAT can consider the response as a graded response and refit the machine-learned NLP subcomponent). After Ms. Kelly finishes her review, she hits the "Launch Dialogue" button to send the messages to her students. The messages can be delivered via text, email, or messaging internal to ASSISTments or the platform in which DRIVER-SEAT is embedded (i.e., Google Classroom).

The next day, Ms. Kelly uses the follow-up interface to review the progress of the eight dialogues. For instance, she observes that a student who seemed confused did well on an assigned problem set, so Ms. Kelly relies on the default response, "Glad you got this down." Starting a dialogue is only the beginning of DRIVER-SEAT's capabilities - it also supports the teacher in making sure that most dialogues end quickly and on a positive note. Of the eight dialogues, Ms. Kelly chose to conclude seven with short messages acknowledging she had reviewed students' actions and marked them satisfactory. One of the students had been assigned a Skill Builder which he tried to complete, but failed to show mastery of the standard. Ms. Kelly decides to follow up with a response for the student to come see her after school for extra help. She ends her five-minute response routine with a feeling of relief - she is attending to students' individual needs but she is not overwhelmed by a pile of time-consuming data to process. This also allows her ample time in the classroom to spend on instruction rather than homework review.

7.3.2 Project Activities

The vignette of Ms. Kelly proposes the final version of DRIVER-SEAT. We will employ an iterative design process to create this tool. In Year 1, our team of five development teachers will be compensated for the time they spend helping us design

the dialogue-initiation interface, as well as seeding the system with the desired student contexts, messages, and actions. Each three-month span within the first year of development will be spent focusing on a different part of the process. Each of these steps will conceptualize the problem as one that can be solved by standard supervised machine learning; for supervised learning, we need datasets that include the messages teachers send, as well as negative examples comprised of the messages that teachers choose not to send.

We will iteratively develop a better interface and build a dataset to be used for system training. This will be a hierarchical task where we first answer, "What diagnosis would best be addressed for this student?" before learning, "What action should the teacher ask the student to do?" or "Given a selected diagnosis to address, what is the best way to phrase a message to best engage the student?" Initially, our dependent measure is just to build models that try to accurately predict held-out test data (using standard cross-validation techniques), where the test data is comprised of teacher-created messages they want to send, as well as messages we suggest they send that have been rejected. We will gather additional use data from our pilot teachers and refit our models (adding more data to our semi-supervised modeling framework). The actions that must occur within DRIVER-SEAT are 1) to decide what diagnosis to address for each student with regard to last night's homework; 2) to select the students for whom to suggest messages; and 3) to decide how to word messages addressing the selected diagnosis. Every time a teacher uses DRIVER-SEAT, the dataset grows with the choices carried out by the teacher.

These are the stages for Year 1 through which our five development teachers, all teaching from 7th-grade EngageNY, will be guided. This work will inform the creation of the final product.

Stage 1. Aug, Sep, Oct: We will meet regularly and have the teachers assess the

complete log files for a particular assignment. They will then create dialogues by using the data.

First, the teachers will be asked to evaluate their students' open-response questions, mark them into common error categories, and come up with a message for each student. Each EngageNY question might have a few common-wrong-answer patterns and we want to identify them. To help us do this, each teacher, in addition to grading their students' open-response questions, will also do the same for a smaller sample of responses from others teachers' classrooms. This will help us get diversity in our understanding of common wrong answers and the ways teachers choose to respond.

Next, the teachers will be given the complete set of clickstream data. They will then create, from scratch, the "dialogue start" using the template in the center of Figure SD3. In addition, we will ask the teachers to give a justification for their message, with respect to the data from the detectors. We expect teachers will be able to create 20 of these dialogue starts per day. With 60 school days and five teachers, that will result in a total of 6,000 dialogue starts. During this time, our team will observe these dialogue starts in conjunction with their justifications and analyze patterns. As the teachers find similarities they will begin to break them up by diagnosis (student gaming, student confused, etc.). This stage will result in the WPI team learning to pay attention to specific diagnoses.

Stage 2. Nov, Dec, Jan: We will start to guess, using the information in Stage 1, what diagnosis would be best for the teacher to dialogue with a student based on which diagnoses were most focused over the justified dialogue start (see Figure SD4). We will use the context of the students' actions, along with the 6,000 dialogue starts created in Stage 1, to offer the teachers three choices of diagnoses. We will then learn from the selections they make. We expect they will initially approve of

Participants	#	Year 1	Year 2	Year 3
Development Teachers	5	- Chosen from group of pilot teachers at the beginning of Year 1 based on volition and motivation - Train with Lesley University Coach to understand the dialogue-initiation interface and the goals and expectations of their role in Year 2	- Spend approximately 180 hours in Year 2 providing content for the dialogue-initiation interface, writing feedback messages to grow the database and refining piloted messages - Lesley University Coach will work with the WPI team and development teachers after each coaching session with the pilot teachers to inform the team of aspects that are working and parts that need adjustment - WPI and development teachers then refine messages based on coaching feedback	- Spend approximately 180 hours writing feedback messages to grow the database and refining piloted messages - Present at a Cyberlearning Workshop titled, 'Using Machine Learning to Personalize Math Instruction' that will be open and free to middle school math educators interested in attending. The focus will be on sharing the results of the work and encouraging more teachers to engage in Cyberlearning within their classrooms
Pilot Teachers	15	- Learn to use ASSISTments through trainings, coaching, and experimentation	- Use the DRIVER-SEAT tool with newly developed feedback options from the five development teachers	- Use DRIVER-SEAT - Meet with Lesley University Coach once a month for one hour each week to receive guidance
District Math Leaders	5	- Train with the pilot teachers to learn how to use ASSISTments to support math instruction by the pilot teachers	- Spend eight hours throughout Year 2 of the grant continuing to oversee the usage of the program in their pilot teachers' classrooms and to provide support, as needed, to assist in pilot teachers using the program consistently and effectively	- Continue to help us as research partners

Table 7.2: Participant timeline.

about 50% of the diagnoses we select, which over time will improve to 80% or 90%.

Stage 3. Feb, March, April: After six months, we will have collected data (12,000 instances) on the type of actions the teachers assign (see Figure SD5). We will use this information to begin to suggest actions from which the teachers can choose. The selections that teachers make will help us improve action suggestions.

Stage 4. May, June, July: During these months, we will focus on creating just the right message for teachers to select in the dialogue-initiation interface (see Figure SD2). By the end of the first year, we will have a working interface to use with the pilot teachers in Year 2. We want to begin with diagnosis selection because it will greatly narrow the options for the actions and messages moving forward.

Timeline for participants

Each year, the team from Lesley University will hold training sessions and workshops for participants. WPI and Lesley University will answer the research questions laid out in section C.4 and will disseminate findings through publications in peer reviewed and prestigious conference and journal venues.

7.3.3 Details of the system behind DRIVER-SEAT

DRIVER-SEAT depends on two main functions to work, the student-context builder and the dialogue-builder. The dialogue-builder relies on the student-context builder to create the options from which the teacher selects a feedback message. The data pipeline begins with the teachers' selection of problems and includes students' responses to previous and targeted assignments from ASSISTments as well as their response to the previous dialogues delivered by DRIVER-SEAT. Once the two builders have made their decisions, the information is shown to the teacher through the dialogue-initiation interface, the context-report, and the follow-up interface.

The student-context builder

The primary function of the student-context builder is to interpret and summarize raw student data and address the data-fusion problem that exists regarding the multiple detectors and data sources. The role of this partition is to build the "student context," defined as the collection of information that is currently available for a particular student, considering the most recent homework assigned to the student, the history of that student interacting with the learning system, and the history of dialogues opened between the teacher and student through DRIVER-SEAT. These multiple sources of information must be combined together in a manner that identifies behaviors exhibited by each student.

To do this, the system will leverage the many existing detectors of student performance, behavior and affect (Table 1), and expand beyond these with a set of detectors that consider student text submitted in response to the many open-response questions assigned by the teacher. These detectors applied to student open-responses will employ techniques drawn from the study of NLP in order to generate estimates of student performance and understanding from their submitted text. They will be

referred to hereafter as NLP-detectors.

In developing the NLP-detectors, the goal is not to automate the grading of such responses, but instead, to estimate how much effort students put into their answers; this further coincides with the content of EngageNY and other OER as the questions commonly ask students to explain or justify an answer in their own words. In this regard, the problem differs from traditional essay-grading tasks in that the student is not evaluated on grammar and structure, but rather on effort and understanding. Similar to previous works exploring the automation of essay grading, the NLP-detectors will utilize a deep recurrent memory network (RMN) (Tran, Bisazza, & Monz, 2016) to assess student understanding for both its ability to leverage the sequential nature and word ordering of text to inform estimates of a dependent measure, but also for its ability to utilize a pool of labeled example responses. As teachers are able to identify student responses that exhibit understanding and effort, as well as those lacking in such measures, the development teachers will be able to supply examples of acceptable and unacceptable responses on which to train and build the NLP-detectors. With such examples, the detectors could further leverage not only the student data from the development teachers, but also all data pertaining to such problem types in EngageNY.

Considering the large number of available detectors of student performance, behavior, and affect (see image 4), in addition to the NLP-detectors and the knowledge of changes in behavior and performance over time, there is one final aspect that must be included in the development of the student-context builder.

Dialogue builder

The dialogue-builder uses the student information to generate the messages and actions that are suggested to teachers. It is an iterative process that aims to learn how

Type of Diagnosis	Action	Description / Expectation
Any	Assign problem set	Assign at least one problem for the student to complete. It could be a Skill Builder.
	Tell me what happened	The student is expected to reply with an explanation for the detected performance/behavior/affect.
	See me	Define a time to meet with the student (e.g. after class).
	Send Content	Have the student watch a video of how to do the homework assignment
	No Action	Assign no action to the student.
Behavior-Specific	Stop Behavior	The student is expected to stop the detected behavior (e.g. gaming the system) on the next assignment.
	Motivation Video	Ask students to watch a motivational video.
Completion-Specific	Finish Assignment	The student is expected to complete the previous assignment that was left unstarted or incomplete
Open Response-Specific	Revise Explanation	The student is expected to rewrite the open response with more consideration.
	Select the Best Response	The student is presented with three responses and is expected to select the best.

Table 7.3: Potential actions to be utilized by DRIVER-SEAT

to maximize the positive impact of such messages. Information from the student-context builder inform the dialogue-builder. A selection process is required to identify the most impactful messages to present to teachers and also the individual students for which each is detected; essentially, this selection process references the need to identify what to say about each student, if necessary, and what action should accompany that message. In consideration of teacher capacity, impact, and student accountability, it follows that only one diagnosis per student is to be presented to the teacher. Additionally, limiting the number of students for which messages are presented to the teacher will further help direct attention to those students who may benefit most from a dialogue.

In this way, the dialogue-builder will incorporate similar methods as Google's Smart Reply. The system will consider the constructed student contexts and prioritize students who are most likely to benefit from a teacher-provided dialogue. For the selected students, two models will each generate the messages suggested to the

teacher as well as a suggested action; assignable actions, as described in Table 3, are an important part of the teacher-student dialogue. These models will be developed in consideration of research questions described in section C.4.b.

In Year 1, the development teachers will provide more feedback than will be typically expected, thus jump-starting the process. There is an inherent priority that is associated with each student context; it is likely more impactful to focus on such detectors as assignment completion or low effort rather than a detector of, for example, boredom where a single dialogue is less likely to contribute a profound effect. This inherent priority of detectors can be learned by paying attention to the types of messages selected by the development teachers. The described inherent priority further extends to the student selection process. The student history, frequency, and content of previous dialogue, as well as the recent measures identified by the detectors will help identify the students for whom a new dialogue may have the most positive impact.

The user interface of DRIVER-SEAT

The first step of DRIVER-SEAT is to support the creation of a dialogue between teacher and student that is rooted in the context of the students and their performance in ASSISTments. The first product is the dialogue-start, as shown in Figure SD3. This is made up of a reference to the context information, a message, and an action. Teachers have a volitional role in creating this start by picking and altering the components. They do this through the dialogue-initiation interface sketched out in Figure SD2. Under column 2 there is a "show" button to display the student information driving the diagnosis; there is an example of such log files in the dialogues of Figure SD3 that shows the step-by-step actions taken by the student that informed the diagnosis. This will give a variety of detailed information about what

the detectors saw that landed the student on this list with this diagnosis. There will also be three choices of messages and a pull down for a selection of actions, as well as ways to change those.

The next day, the dialogue continues when the teacher accesses the follow-up interface to check on the student completion of the assigned action and to reply. This interface is not shown, but it will be a vital part of the teacher's routine. If need be, a new dialogue will be launched if the student has not finished the action to the teacher's specifications. Only one open dialogue per student will be permitted in the system to avoid overloading the teacher capacity in maintaining such dialogues and also to avoid overloading the student with assigned tasks. Records will be kept by the student-context builder and the dialogue-builder in order to improve their selections.

7.3.4 Experimentation and Exploration

We plan to evaluate this project through both qualitative and quantitative measures while addressing pertinent questions in the fields of machine learning, computer science, and education by means of the system's development and deployment.

The utility of DRIVER-SEAT is dependent on its use by teachers. As such, the clear question to first address is whether teachers like using the system, and whether they find that it is helpful in initiating meaningful dialogues with students. Using qualitative methods from human-computer interaction this question becomes: does the system meet its goal of supporting data-informed communication with students while considering the limitations of teacher capacity? System usage statistics and feedback from teachers using the system during its development will serve as measures for the evaluation of this goal. We will utilize self-report surveys, think-aloud protocols, and other feedback gathered from development teachers and pilot teach-

ers to understand which aspects of the system are most supportive, as well as those that need improvement. These measures will also act to iteratively improve the utility of the system throughout its development.

Aside from the qualitative measures described above, quantitative evaluations can be gained through small-scale randomized controlled experiments (RCEs) conducted within the system, as described in the subsections below.

7.3.5 Research Questions Addressing Issues in Human Learning

How do teachers' capacities for tailoring feedback messages and students' perceptions of message origin alter the effects of feedback messages on subsequent student performance?

A randomized controlled trial to evaluate issues within the efficacy of implementing DRIVER-SEAT will be conducted at the end of Year 1, with replications intended for the ends of Years 2 and 3. The goal of this experiment will be to assess how student performance outcomes differ on the next night's homework following receipt of feedback messages crafted by teachers under a variety of circumstances. The five development teachers will come together at our workshops, each with approximately 100 students, creating a pool of 500 students. Student data will be anonymized and randomly assigned to teachers. This stratification will remove potential biases that could otherwise arise among individual teachers' students or feedback styles. Each teacher will then be asked to provide feedback to 125 students. Random subsets of 25 students will be generated and randomly assigned to receive one of five conditions of feedback described in Table 4 below. These conditions differ based on whether or not the teacher utilizes DRIVER-SEAT, the amount of time the teacher

		Feedback Creation Method	
		DRIVER-SEAT	Traditional
Allowed	Infinite	Assistive	Laborious X Teacher ——————— Laborious X Computer
	Short Period	Efficient	Unrealistic

Table 7.4: Conditions within the randomized controlled trial evaluating feedback development method, time allowed, and students' perceptions of message origin.

has to provide feedback to individual students, and whether students are made to perceive messages as originating from the teacher or from the computer system. We expect that when given an infinite amount of time while using DRIVER-SEAT, the computer will play an assistive role in feedback selection (Assistive). When DRIVER-SEAT is paired with short periods of time for feedback selection (i.e. one minute per student), which we suspect will be the ideal use-case of DRIVER-SEAT, we expect high efficiency (Efficient). We expect that when asked to apply traditional feedback creation methods using ASSISTments reports (i.e., Ms. Kelly's vignette in section C.1), teachers will feel overwhelmed and fail to address the concerns of each student (Unrealistic). When asked to apply traditional feedback creation methods without time constraint, we expect that teachers will find the task daunting, but possible (Laborious). Subsets of the Laborious condition will examine the effect of students' perceptions of message origins, as hailing from the teacher or from the computer. These subgroups were established because we suspect that student performance will increase when feedback messages are perceived as being penned by their teacher.

Analysis of students' performance on the next night's homework will allow for a series of pairwise comparisons to isolate:

- The main effect of DRIVER-SEAT in preparing feedback messages.

  Comparing messages crafted within Assistive and Efficient conditions with

those crafted within Laborious and Unrealistic conditions will allow us to assess whether DRIVER-SEAT helps teachers select appropriate and effective feedback messages for students. We suspect that messages sent using DRIVER-SEAT will be no different than messages sent using traditional methods, but that they will be selected more efficiently, allowing teachers to reach a greater number of students and to better understand the issues arising in their students' open responses.

- The main effect of time allowed in selecting or creating feedback messages.

  Comparing messages crafted within Assistive and Laborious conditions with those crafted within Efficient and Unrealistic conditions will allow us to assess whether student performance is impacted by the amount of time that teachers utilize when writing or selecting feedback. In current online learning environments, teachers face the problem of limited time to spread across a high quantity of students requiring feedback. We suspect that DRIVER-SEAT will be helpful because of its efficiency, and that short time periods will allow teachers to provide communication that is more instructionally relevant.

- The interaction effect of DRIVER-SEAT and time allowed.

  Looking across all conditions with a focus on the comparison between Efficient and Unrealistic, we hope to show that messages created under duress without the assistance of suggestions from DRIVER-SEAT will be of lower quality and less effective in producing learning outcomes. We speculate that teachers working under a time constraint will be more likely to focus on summative measures (i.e., "You got 85% correct, Not bad!") when more substantive responses are more likely to affect change in students' subsequent performance (i.e., "You consistently had trouble adding negatives, let's talk

about how to approach these problems!").

- The main effect of students' perceptions of message origin.

  Comparing messages crafted within the Laborious condition subgroups, in which students are made to perceive messages as originating from the teacher or the computer will allow us to assess the importance the perceived role of the teacher in personalization, in an idealized setting, free of time constraints. We suspect that messages perceived as originating from the teacher will increase student performance through measures of affect and engagement.

What is the short-term effect of DRIVER-SEAT under realistic conditions?

To what extent, if any, does the use of DRIVER-SEAT alter the behaviors exhibited by students and increase student learning under real-life conditions? To address this question, we will conduct a field trial in pilot teachers' classrooms. After each assignment, the DRIVER-SEAT detectors will search for problematic behavior among all students, but display only a random subset of the results to the teacher. So each morning the system will display up to eight students, so as not to overload the teachers. We will compare achievement on the subsequent assignment between those randomly-selected students displayed and those detected to be exhibiting similar behavior but not randomly assigned to be be displayed on DRIVER-SEAT. A statistical analysis pooling year-long data, and accounting for the longitudinal design by controlling for the numbers of messages students received previously and clustering standard errors at the student level, will estimate overall average effects of the DRIVER-SEAT display.

7.3.6 Research Questions Addressing Issues in Computer Science

A number of challenges emerge in regard to the development of DRIVER-SEAT and the infrastructure needed to support its functionality. Particularly in consideration of the methodologies comprising Google's Smart Reply system (Kannan et al., 2016), the differences in applied fields lead to several research questions surrounding the identified parallels exhibited in the systems. It is without question that the infrastructure of DRIVER-SEAT will need to utilize a number of techniques from computer science and machine learning, in addition to the exploration of NLP methods beyond even those applied within Smart Reply.

Do different representations of student data significantly impact model performance when generating messages?

Perhaps the largest difference between DRIVER-SEAT and Smart Reply is the structure of data being presented to the system. Smart Reply focuses its attention purely on the natural language of incoming emails. The data that is coming from the educational system feeding DRIVER-SEAT, however, contains a mixture of strongly and weakly structured student information, contained within the student context. The student actions, summarized by the detectors of performance, behavior, and affect, combined with the information pertaining to student history and answers to open-response questions comprise different sources, or channels, of information that must be considered simultaneously by the student-context builder in order for the system to decide what to address for each student.

The manner in which these different channels of student information are represented to construct the student context may vastly affect the ability of DRIVER-

SEAT to generate viable messages. We have explored in previous work (Zhang, Xiong, Zhao, Botelho, & Heffernan, 2017) how to combine several channels of student information into a single model, and found that the representation had a significant impact on model performance. Utilizing another deep learning technique, we applied an autoencoder (Rumelhart, Hinton, & Williams, 1985) that learns a lower-dimensional rich feature embedding that describes a set of inputs. This lower-dimensional representation was found to be helpful when attempting to input several channels of student information into a recurrent deep learning model. Here, we will explore if the representation exhibits the same utility, comparing this method against more traditional representations (i.e., directly feeding the channels into the message-generation model).

Does the use of a Recurrent Memory Network lead to improvements over the LSTM and clustering methodology employed by Smart Reply?

Google's Smart Reply system employs a multi-step process to generate and select email messages to suggest to users. Within this, they are using clusters of messages with labeled semantic intent to estimate the meaning of each generated message; this helps both to validate that the message is appropriate and also to help ensure diversity amongst the suggested messages. The clustering of messages allowed for the application of interpretable, human-coded labels to be applied to the generated messages. However, to gain an understanding that two messages are different from each other, such a human-readable label is not required and may be better modeled by a different, deep learning model known as a Recurrent Memory Network (RMN) (Tran, Bisazza, & Monz, 2016).

RMNs are a type of deep learning model that has been developed from traditional recurrent neural networks (Williams & Zipser, 1989) to incorporate a static memory

that can be used to produce more data-informed estimates. The static memory used in this type of model is often comprised of a set of embedded example cases that the network is able to use in conjunction with an input sequence. As stated in section B.4, our team has applied RMNs in the past to automatically grade student essays. In that model, a student essay is read word-by-word into the model and compared against graded example essays included in the network's memory; by comparing against the examples, the model estimates the score of the input essay.

In this way, the LSTM and clustering methodology used in Smart Reply and the RMN used for automatic essay grading are performing the same general function. Each compares an input sequence against a set of examples to select a set of messages to suggest. Applying the RMN could help simplify the workflow by incorporating the message generation and selection into the same step, unlike as it is done in Smart Reply. Comparing the two methods will identify if 1) the Smart Reply methodology appropriates to education with comparable accuracy and reliability and 2) if the RMN exhibits better performance despite the inability to observe human-interpretable semantic intents.

To what extent can we reliably incorporate specificity into the messages, particularly in considering student open-responses?

As described in the Problem section, many computer-based systems fail to consider student open-response answers in evaluating student performance despite the NLP techniques that exist to aid in interpreting such text. It is uncertain, however, the degree in which specific aspects of the open-response problem in conjunction with the student's response can be used to build suggested messages. The development of the NLP detectors thus far have described how we can use deep learning models to estimate understanding and effort, but we also intend to explore how incorporat-

ing more information may provide an even more prominent role for student open responses in the message creation.

A benefit of observing student performance within a commonly used OER, such as EngageNY, is that many students are writing answers to the same content, providing the opportunity to observe various different correct and commonly incorrect responses. The ability to identify these common incorrect approaches from the student texts could help drive more specific suggested messages that teachers would want to say to address the recognized misunderstanding. This specificity is illustrated in Figure SD2 by Wei, Wakeeta, and Linda, where the system has estimated not only that the open-response was incorrect, but was able to suggest why this was the case to the teacher and provide three suggested messages addressing this recognized behavior. Using a similar methodology as Smart Reply, we will explore how to group and interpret the semantic intent of student open responses for inclusion into the suggested messages.

7.4 Broader Impacts

As the reader can see, this grant will pave the way for critical research to answer important research questions from both computer science and human learning. The findings from the research questions will help advance the fields of computer science, as well as advise our understanding of the types of messages that can most effectively promote student learning. The broader impacts of this work will to help thousands of teachers more efficiently communicate with and provide feedback to their students to improve learning. Supporting teachers to provide direct and supportive feedback to students helps promote a sense that the teacher is paying attention and cares about student progress; such support could be transformative.

Chapter 8

The HAND-RAISE Intervention through LIVE-CHART: Directing Teachers' Attention to Prevent the Loss of Student Interest in STEM

The following grant proposal was written alongside PI's Neil Heffernan and Korinn Ostrow. This proposal is pending submission. Supplementary figures and materials as well as references for this proposal are included as appendices at the end of this dissertation document.

8.1 The Problem

The United States is desperately interested in transitioning it's workers into critical jobs in STEM-related fields, but many students lack interest and the necessary skill sets in mathematics and sciences upon leaving school. As a result, fields that have

always depended heavily on math (e.g., physics and chemistry), and others that are evolving to require more and more math (i.e., biological sciences), are undoubtedly affected by student's declining attention to core fields within mathematics and science.

Declining interest in STEM education is well-documented. Measures of students' interest in STEM-related subjects commonly show high interest in elementary school that recedes with each year as students advance through middle and high school. By the time students graduate from high school, many have completely lost the innate interest in STEM that existed in childhood (Potvin and Hasni, 2014; Mahoney, 2010). This decline is observed across grade levels within the United States (Alexander et al., 2012; Gottfried et al., 2009; Sorge, 2007; George, 2006) as well as internationally (Potvin & Hasni, 2014; OECD, 2006; Sjberg, & Schreiner, 2005; Osborne and Dillon, 2008).

Explanations have been offered in previous works to attempt to explain why such a decline is observed, attributing the loss of interest to the poor framing of subject matter as practical or relevant (Barmby, Kind, & Jones, 2008), due to a higher focus on standardized testing (Guvercin, Tekkaya, &, Sungur, 2010), or even due to the quality of instruction (Krapp & Prenzel, 2011). However, another theory suggests that students' self-concept (i.e. their perspectives of what they know and their confidence in the subject matter), the attitudes of their peers, and the quality of student-teacher interactions may explain some of the decline (George, 2006). This theory largely suggests that daily interactions occurring in the classroom can lead to a gradual loss of interest. There is also a significant and growing problem in our culture by which female and minority students express less interest in STEM careers than their male counterparts. We believe that much of this discrepancy emerges from disillusionment or frustration in math classes and the way that teachers interact with

their students individually and as a group. On average, females and minorities are documented as exhibiting lower participation and engagement in classroom activities than their peers (Greenfield, 1997; Bernacki et al., 2016).

We seek to address this problem using an intervention aimed at building student confidence and self-concept and promoting more informed, quality interactions between teachers and their students to support positive help-seeking and answering strategies in the classroom. Using a computer-based intervention, we will provide teachers with a set of tools to augment their ability to pay attention to students in their class, specifically directing them to attend to students who would benefit most from assistance. These tools, collectively called the HAND-RAISE Intervention, will also help students develop the skill set required to articulate questions while building their confidence and providing opportunities to engage in peer support.

Students often refuse to raise their hands in class due to a lack of confidence, math anxiety, or a diminished feeling of belonging with relation to their peers. We believe that these aspects align directly with the most prominent documented causes of decline of interest and participation in STEM-related subjects and, later, careers. We do not suggest that our proposed intervention will renew lost interest in STEM. Instead, we hope that our tool preempts the decline by maintaining students' interest over the course of critical school years by focusing on facilitating higher student engagement and fostering high quality student-teacher interactions within the classroom.

8.2 The Opportunity

The previous section highlighted a set of problems pertaining to students' engagement and interest in STEM. In this section, we will identify several opportunities

that can be leveraged in the development of the aforementioned HAND-RAISE Intervention with particular focus on the online learning platform ASSISTments through which we plan to develop and deploy the intervention.

8.2.1 Results of Prior NSF Support: Heffernan's ASSISTments and other CoPIs

ASSISTments.org was created while conducting research for past NSF awards. Heffernan's NSF CAREER award (CAREER: Learning about Learning award 0448319, $646,075, 2006 2013) is the most relevant grant that helped to create ASSISTments. It's intellectual merit included more than four dozen peer-reviewed publications in machine learning, deep learning, clustering, prediction, etc. (see the separate section in the references noting these 60+ papers). Other NSF grants have also supported ASSISTments, including NSF 0742503 whose intellectual merit included 22 published randomized controlled trials measuring different ways to improve student learning through feedback (see the separate section in the references noting these 22 papers). The broader impact of these awards has been support for thousands of students via the ASSISTments service provided as a free public service by WPI. Last year alone, students solved more than 12 million problems. Over more than a decade, the WPI team has written tens of thousands of questions and teachers have written an additional 75,000 questions for their students. The ASSISTments platform allows teachers to enhance any homework assignment with online feedback for students and reports for teachers (see Figure 1 for assignment questions and the associated item-report). When students solve a problem in ASSISTments, after entering their answer they are told if they were correct. If they got the problem wrong, they can try again, and in some instances, receive additional tutoring. The element of data entry allows for the creation of class and student reports that teachers can

use to inform the next day's instruction. The item-report is designed to provide teachers with information that is easy to respond to. To explain further, we provide a vignette of Ms. Kelly, a 7th-grade math teacher (an actual video of Ms. Kelly reviewing an item-report can be accessed at Kelly et al., 2013). Let us assume Ms. Kelly assigned the problems in Figure 1 to her students using ASSISTments. The next morning, Ms. Kelly prepares for class by assessing the item-report. She would probably want to talk to her students about question 1, as it was challenging: only 27% of students provided the correct answer and 66% of students who got it wrong responded with an answer of '0'. She realizes that these students made a common error of subtracting -9 from 9 to get zero rather than solving 9 - (-9) correctly to reach 18. She decides to spend a portion of her class time addressing this misconception. She then reads through some of the open- response answers and notices that many students, including Grace and Billy, thought that Mountain Charter was always better, failing to see how one plan was better for more people while one was better for fewer. She also decides to spend a portion of her class time addressing this misconception.

Figure 1: An example of an ASSISTments Item Report

Although the item-report is an exceptionally helpful tool for teachers, as shown by Ms. Kelly's vignette, teachers commonly fail to look at these reports until after students have completed their assignments (e.g. before the class period on the subsequent day or for grading purposes). As such, it does not provide teachers with the opportunity to intervene in real-time to help students on classwork, and does not help to direct teachers to the questions that arise as students are actively working. In our solution section, we will describe how we plan to provide teachers with such tools to help support real-time action within the classroom.

SRI study that showed ASSISTments caused better learning and closed achievement gaps

An in-depth randomized controlled trial conducted by SRI International recently concluded (Roschelle, Feng, Murphy, & Mason, 2016), producing three main findings: 1) Teachers reliably changed the way they reviewed homework, consistent with the intended-use model. They still spent approximately the same amount of time reviewing homework, but their reviews were focused on a smaller number of difficult problems rather than all questions. 2) Students in schools randomly assigned to use ASSISTments had reliably higher rates of learning (p¡.008) as shown by an additional eight-point gain on an end-of-year standardized test, an approximate 75% gain atop the 11 points students are expected to gain (on average) in a school year. 3) The intervention helped to close achievement gaps, whereas Steenbergen-Hu and Cooper (2013) found that most other K-12 mathematics intelligent tutoring systems instead exacerbate this problem. Students with incoming 6th-grade scores below the median experienced greater gains than those above the median.

Helping students through peer assistance

Another set of tools within the ASSISTments learning platform are aimed at facilitating peer assistance for students working on the same content (e.g. peers within the same class). This tool, called PeerASSIST (Selent, 2017), collects student work and explanations as students work through an assignment, which is then used as computer-provided aid to other students who may be in need of help to solve the same content. Essentially, if students know how to solve a problem and are able to articulate how they arrived at the solution, such information could be helpful to other students when they are struggling.

From the basis of PeerASSIST, a function called "Star Student" was then im-

plemented. Star Student works with peerASSIST along with teacher settings that allow the teacher to deem a student as an exemplary student to provide assistance to peers. Once a student has been deemed a "Star Student," work and explanations created by that student will be distributed to other students in need of assistance on the content through PeerASSIST. This not only helps provide students with aid, but also can help build students' confidence knowing that the teacher recognizes their work as being exemplary. These ideas will be further implemented by the intervention proposed in this project, with further detail provided in the solution section.

8.2.2 Opportunities through AI-Enhanced Classrooms

Recent work by Ken Holstein recognizes the need for real-time tools in the classroom to promote student-teacher interactions (Holstein, McLaren, & Aleven, 2017; Holstein et al, 2018). In that work, the authors describe the implementation of mixed-reality glasses worn by the teacher to provide real-time notifications of student performance and behavior in the classroom. Teachers wearing the device are able to see notifications on specific students in the class as well as have the ability to pull up recent activity for individual students as they work in a computer-based system. The goal of that work was to direct teachers' attention to the students who may most benefit from teacher attention. A study found that through the real-time notifications, teachers spent more time with students with lower pretest scores as compared with when the glasses were not used.

This work represents several great opportunities in that it demonstrates the benefits of providing teachers with real-time notifications of student behavior and performance to direct attention to those who may most benefit from an interaction. It is difficult and somewhat impractical, largely due to financial constraints, to

supply teachers with mixed-reality glasses, but by connecting a set of tools that can be used through a tablet or even desktop computer (i.e. through any device with an internet browser), we can provide teachers with the same benefits. Furthermore, as the intervention proposed in this project is to be delivered through ASSISTments, it can interface directly with the online learning platform to connect teachers with the student data already collected within the system.

8.2.3 ASSISTments currently has detectors that rely on student input

In order to develop a tool that is able to better inform teacher-student interactions, it is important to provide useful notifications of student behavior and performance during the class period. In recent Holstein et al.'s work (Holstein, McLaren, & Aleven, 2018), the device focused on a small set of behaviors on which to notify teachers. These included, for example, when a student appeared idle, or appeared to be gaming the system (Paquette & Baker, 2017). While the behaviors used in that work are certainly reasonable, other detectors have been developed using ASSISTments data in the past and may be explored for use in this project. Such detectors of student affect (Ocumpaugh et al. 2014; Botelho et al. 2017) as well as other constructs such as carelessness (San Pedro et al, 2014) have been developed using ASSISTments data and have even been shown to be predictive of student involvement in STEM-related majors in college (San Pedro et al, 2014) and whether students will pursue STEM-related careers (Makhlouf & Mine, 2018).

These, in addition to other notable detectors such as those of student gaming (Paquette & Baker, 2017) can be utilized to explore the utility of reporting such measures to teachers in real-time during the class period.

8.3 The Solution

In this project, we will develop and deploy an intervention aimed at increasing student interest and engagement in STEM-related subjects by supporting students in the development of positive help-seeking strategies as well as build confidence to both ask and respond to questions in a classroom setting. To accomplish this goal, we will develop a set of teacher- and student-facing tools to help facilitate engagement during class periods in real time. These tools are described through the development of two tools: HAND-RAISE and LIVE-CHART.

Live Interactive Visual Environment for Creating Heightened Awareness and Responsiveness for Teachers (LIVE-CHART) seeks to provide real time notifications of student performance as they work on classwork, while Help-seeking Application for Notifying and Driving Real-time Actions to Increase Student Engagement (HAND-RAISE) provides the means for students to ask questions through a tool that helps develop positive help-seeking behavior while allowing teachers to address questions in more efficient ways. Together, these tools describe the HAND-RAISE Intervention that will be developed and evaluated through the proposed project.

The following sections will provide a vignette to exemplify some of the planned functions of these tools as well as other details regarding the planned development and evaluation processes. While the tool itself is not limited to middle school mathematics, we plan to focus the development and evaluation methods on seventh grade mathematics as it is a grade level where many core mathematical concepts are introduced (such as algebra and equation solving) that are integral to subsequent STEM-related subjects while simultaneously targeting the optimal age range for decline in STEM interest.

Figure 2: The LIVE-CHART classroom display interface.

8.3.1 The HAND-RAISE Intervention Vignette

This vignette will detail a hypothetical use case of the HAND-RAISE Intervention delivered through LIVE-CHART in order to demonstrate what we describe as the final version of both of these tools. Coinciding with Figures 2, 3, and 4, this vignette follows an example teacher, Ms. Kelly, as she addresses her 7th grade class of 8 students using ASSISTments to complete their classwork on equation solving.

To begin, Ms. Kelly logs in to her ASSISTments account using her tablet device and navigates to the LIVE-CHART tool to view her students; while she knows she has the ability to use LIVE-CHART through her desktop computer as well, she uses the tablet as it allows her to stay connected as she moves around the classroom while assisting students. She first sees that everyone has logged in to ASSISTments with the exception of Logged-off Logan who she has already noted is absent. This assures her that students are ready to work on their ASSISTments work and are on-task at the beginning of the class period. Over the course of the class period, as students are working through problems within the system, Ms. Kelly will be notified of particular events that she is likely to want to address. Figure 2 illustrates a more extreme example of such notifications, where 5 of her students are exhibiting behaviors for which she should direct her attention.

Figure 3: The LIVE-CHART student display to illustrate the data made available to teachers through the interface.

Ms. Kelly first addresses Getting it Grace. She clicks on the notification to out more information about Grace's performance, leading her to a screen illustrated by Figure 3. On this screen, Ms. Kelly can clearly see that Grace was struggling to learn the material early in the assignment, but is now beginning to answer problems correctly indicating that she has learned the topic. Ms. Kelly walks over to Grace and tells her what a great job she is doing and decides to make her a Star Student

for the topic of equation solving; she selects the option to "star" Grace through the LIVE-CHART student display, meaning that Grace will now be able to help answer the questions of other students on the identified topic via PeerASSIST. Ms. Kelly then closes the notification indicating that she has addressed that Grace is doing well, and the icon next to Grace's avatar disappears.

Ms. Kelly looks again at the LIVE-CHART display of her classroom and notices that the "idle" notification next to Bored Billy has disappeared. Ms. Kelly has set up her LIVE-CHART to reflect the true seating chart of her classroom, so when she spoke to Grace, she happened to be standing near Billy who was seemingly off task. When Ms. Kelly addressed Grace, Billy directed his attention back to his work, causing the icon to disappear. While Ms. Kelly could still address Billy's prior off-task behavior, she instead decides to give him a chance to remain engaged, as she knows his focus has returned to his work. She makes a mental note to check LIVE-CHART again in a few minutes to ensure that he remains on task.

Returning once again to the classroom display, Ms. Kelly directs her attention to Hand-raised Henrietta and Hand-raised Harry. She wants to ensure that both Henrietta and Harry get timely assistance. In this regard, Ms. Kelly has several options regarding how to proceed. If she cannot address Harry within 2 minutes, Ms. Kelly has enabled her settings to allow the HAND-RAISE tool to send Harry's question to the highest-recommended student; such a student would need to either be deemed a Star Student for the topic, or have correctly solved the particular problem on which Harry is currently working or successfully completed the assignment (if it is a mastery-based assignment). Similarly, Ms. Kelly could select Harry's notification, which changes the LIVE-CHART display to indicate all recommended students (based on the previously described criteria), and drag the notification to effectively send the question to the selected student. Lastly, Ms. Kelly could of course

address Harry's question herself, but as her attention is first drawn to Henrietta's question, she allows the HAND-RAISE tool to let Grace answer Harry's question, as she had just selected Grace as a Star Student for the topic.

Grace is notified through ASSISTments that a question has been directed to her and she accepts the request (other possible options will be detailed further in Section C.1.a), and reads the question from Harry. She then writes a response, describing where she believes Harry is becoming confused, and takes a picture of her notes to include with her response.

While this is occurring, Ms. Kelly begins to address Henrietta's question. Ms. Kelly first selects Henrietta's avatar from the LIVE-CHART classroom display, showing her Henrietta's recent performance in addition to the question she has asked and the problem on which she is currently working. Ms. Kelly selects the icon next to the displayed question to indicate that she is addressing the issue herself and begins to talk with Henrietta. By selecting the question, the LIVE-CHART display switches to a scratch pad and, as she already knows Henrietta's specific question, Ms. Kelly is able to write out a worked example to help clear up her confusion. Once finished with the example, Ms. Kelly indicates that she is done and an image of the scratch pad is saved and sent to Henrietta to use as a reference as she continues to work through the problem; alternatively, Ms. Kelly could have typed the example out or written the example on paper and used her tablet to take a picture of the work to send as well (or she could have discarded the example if she felt that it would be unhelpful to Henrietta). Upon returning to the LIVE-CHART classroom display, the HAND-RAISE icon for Henrietta disappears as her question was sufficiently answered and subsequently Harry's icon disappears as well because Grace was able to answer his question sufficiently.

Figure 4: The HAND-RAISE interface from the perspective of the student work-

ing in ASSISTments.

This leaves only one icon remaining for Ms. Kelly to address, and that is the "possible gaming" notification next to Gaming Ganji. Ms. Kelly again uses the student information display by clicking on Ganji's avatar. Ms. Kelly sees that Ganji has been asking for a lot of hints from ASSISTments very quickly, causing LIVE-CHART to believe that he is not using the hints to learn and is instead attempting to "game the system." Ms. Kelly, knowing Ganji well having had him as a student throughout the year, infers that he may be confused and is reluctant to ask a question. As such, Ms. Kelly selects an action on the student information display which sends a message to Ganji asking him to articulate a question that he may have. Ganji responds to his teacher's prompt by describing his confusion to the best of his ability which is then sent back to Ms. Kelly. With a clearer idea as to what is giving Ganji trouble, Ms. Kelly opens the scratch pad and approaches Ganji to offer a worked example to address his confusion. Ms. Kelly reminds Ganji that he should ask a question and should utilize the HAND-RAISE tool the next time he is having trouble that available hints are unable to remedy (rather than exhibiting gaming behavior).

The functionality of HAND-RAISE and LIVE CHART

The vignette of Ms. Kelly offers a description of several displays and functions that comprise both the LIVE-CHART and HAND-RAISE tools. This section will provide a brief overview of some of the planned functions of these tools as they will be displayed to both teachers and students, with larger focus on those aspects that were not able to be highlighted by the vignette.

The most prominent feature of the intervention is that of LIVE-CHART's classroom display. The display itself is meant to provide teachers with a real time view

of students working in the classroom. LIVE-CHART, as a tool offered through ASSISTments, will be able to connect to a teacher's class roster to know which students are in which class periods and provide the necessary tools to allow teachers to edit this roster as is already provided through ASSISTments. In the classroom display, teachers will have the ability to drag and rotate students and props (such as the board and teachers desk as illustrated in Figure 2), so that the layout reflects that of the actual classroom. In this way, the tool helps provide a seating chart-like view so that the teacher may easily find and address students (the notification on a student in the corner of the classroom will correspond with the actual student sitting in the corresponding corner of the classroom.

Selecting a student from the classroom display will open the student display. This display will provide the teacher with data pertaining to the selected student to provide such information as on which problem the student is currently working, the recent actions taken in the system, and other descriptives. In addition to this, when the student has an active notification (such as a hand raised), this display will provide further information about the notification (e.g. the specific question of the student) and provide the teacher with a set of possible actions. These actions will be determined during the initial development process described further in the next section, but it is likely that such actions may be sending a message to a student, indicating that the student was addressed, opening a scratch pad to illustrate an example, or even simply dismissing the notification.

The types of notifications, as will be described further in the next section, will be driven by what emerges as most important amongst the recruited Development Teachers who will aid in the development process of the tools. Such notifications will certainly include an indication of a student raising their hand, but others may likely include indications of student gaming the system (e.g. abusing hints), being idle for

an extended period, and also more positive behaviors such as correctly answering problems after struggling to learn a topic.

HAND-RAISE, from the perspective of the student, provides a means of asking a question to be answered by the teacher or a peer. Students with low confidence may be unwilling to physically raise their hands to ask questions as it may draw unwanted attention, but the HAND-RAISE tool allows students to do so in a more comfortable manner. The student can simply click the HAND-RAISE button from the ASSISTments tutor, which then requires that student to articulate a question; this prompt not only helps the teacher in that the question can appear with the notification on the LIVE-CHART student display, but it also helps students develop the skill set to articulate questions when the material is confusing or difficult.

The teacher may decide to send a student's question to a recommended peer (as was performed in the vignette), in which case it is sent anonymously to the selected peer as a message. The peer is notified of the question and is given several options including the ability to accept (and would then subsequently write a response), but will also include the ability to pass on the question if the solution is not known or if the student is unwilling to answer the question at the given time; it seems unreasonable to require a student to answer a peer's question if he/she is unable or for any other reason, supporting the inclusion of the option to pass. These and additional options will be discussed with the Development Teachers to correspond with identified use cases of interest.

8.3.2 Project Activities

The project is aimed to be developed using an iterative design process guided through the communication and interaction with the Development Teachers through the first two years of the grant. These teachers will play an integral role in develop-

ing LIVE-CHART and HAND-RAISE to best augment their teaching strategies and helping to promote positive help-seeking and responding behaviors amongst their students. The vignette of Ms. Kelly interacting with her students in a hypothetical setting describes our initial design and use cases for the tool; as we described, however, the behaviors for which a teacher is notified as well as the actions made available to teachers will be selected and developed in a data-driven manner gained through interactions and feedback from the Development Teachers.

While we describe the intended timeline for the project and participants in Section C.2.a, we will provide greater detail as to the specific activities planned for the 20 Development Teachers over five three-month stages spanning just beyond the first year of the grant; the three-month timespan illustrates the intended short-term feedback loop intended for the project to promote faster development cycles that are able to effectively incorporate the information gained from the Development Teachers.

Stage 1. Jan, Feb, Mar: The goal of this initial stage is to begin to learn the types of behaviors that are most important to teachers as well as potential actions that can be taken as a result of observed behavior. The development of the LIVE-CHART tool, and subsequently HAND-RAISE, relies on teachers being able to effectively take action to help students become more engaged in the classroom with these tools helping teachers to recognize where such action is needed; this starts, however, with teachers helping to identify cases that are most actionable as well as potential actions that are likely to positively impact student confidence (e.g. being able to praise a student, such as Getting it Grace in the vignette, for doing well, particularly after struggling) as well as engagement (e.g. ensuring students, such as Gaming Ganji and Bored Billy, are on-task and practicing positive learning strategies).

The Development Teachers will spend one hour each night of the week looking at

the action-level clickstream data of their students and identifying what they would address in that data and how they would take action. ASSISTments already provides action-level reports to teachers, but we will provide an augmented version of such a report to the development teachers that includes additional detectors of student behavior (i.e. student gaming), and affective state (i.e. concentration, confusion, frustration, and boredom) to provide teachers with a breadth of information. The protocol that these teachers will be asked to follow is to be guided by a set of informal prompts to help facilitate helpful feedback. Such prompts will include asking the teacher "What, if anything, would you say to the student if he/she were present after looking at the data?" and "Is there any instance where you would praise the student for their work? Where?" as well as other such questions that will evolve as we gain more information from teachers.

Stage 2. Apr, May, Jun: During the second stage, the Development Teachers will continue to look at their students' clickstream data and providing feedback on a nightly basis to continually help inform the types of behaviors that 1) commonly emerge, 2) are commonly identified as important, and 3) can be effectively addressed through clear actions. The type of data displayed to teachers during this collection process will be informed by their responses (particularly in response to the informal prompts). Information such as whether or not a student looks bored, for example, may not be as useful as other detectors of student behavior (or perhaps the reverse), in which case we can learn how to prioritize and select the types of student data on which to focus. It is important that LIVE-CHART, as intended as a real-time notification tool, is developed to be very selective of the types of notifications sent to teachers; it is important to not overwhelm the teacher with information about all students (20 simultaneous notifications occurring each second is likely neither useful nor practical for teachers), but also we do not want teachers to be constantly looking

at their device and ignoring what is happening outside the tool in the classroom; the tool should help provide information to teachers when it is most useful without consuming their complete attention.

At this stage in the development of the tool, the interface by which we collect data from teachers will also be updated to more closely resemble what will become the student-level display of LIVE-CHART as illustrated in Figure 3. As it is through such a display that teachers will be able to view and interpret recent student performance through LIVE-CHART in the classroom, we will also ask for feedback regarding the layout, type, and visualization of data to improve on how such data is represented and displayed to teachers.

Stage 3. Jul, Aug, Sep: As most, if not all, Development Teachers will likely not be using ASSISTments with students during summer months, they will be asked to look at past student data to continue to regularly provide feedback during development. By this stage, however, it is also the goal to provide the Development Teachers with an initial version of LIVE-CHART. This initial version will be designed to play back, in real time or at slightly faster speed, student data from a class period from the previous academic year. In this way, the prototype will simulate a real time classroom by playing back pre-recorded student log data and displaying notifications to teachers as if they were present in the class.

Use of this prototype will help to gain feedback on the types and frequency of student notifications through the system, the user interface, and will also be the first chance that the teachers will be able to provide feedback on classroom-level data. By looking at clickstream data of individual students, as was the case in the first two stages, it is likely easy for teachers to find something that is worth addressing and taking action within each student's sequence of actions. By allowing the teacher to select which students to address from a classroom display (and limiting the displayed

actions up to that instant of the simulated class period), we can learn not only which types of notifications are important to teachers, but also when such notifications are important; the temporal information is likely just as important to consider when deciding what to present to teachers (e.g. it is likely unhelpful to notify a teacher of a student behavior multiple times in a short time span, but perhaps there are instances where this would be important).

Stage 4. Oct, Nov, Dec: The final stage of the first year of development is aimed at improving the prototype version of LIVE-CHART to allow for real-time functionality in real classrooms. Following the development cycle of stage 3 and subsequently the feedback gained from the Development Teachers during that time, it is the goal to provide such teachers with a version of the tool that can be used in their real classrooms during the first half of the academic year. The Development Teachers will be asked to use the prototype in their classrooms at least once per week and continue to look at pre-recorded class periods as had been done in stage 3 on nights where the tool had not been used. We will ask the teachers to, on the night following usage of the tool in their classrooms, follow the same procedure of looking at and providing feedback for the tool using pre-recorded class periods, but specifically replaying the class period where the tool had been used on the previous day; this will allow the teacher to provide feedback on the usage of the tool in the classroom, as it is unlikely that the teacher will have sufficient time during the class period to do so.

Stage 5. Jan, Feb, Mar (year 2): It is the goal of development to produce an initial version of the HAND-RAISE tool and begin implementing its functionality within the ASSISTments tutor and LIVE-CHART by the end of this stage. As the HAND-RAISE functionality is a focal point of the intervention described in this project, while facilitated through the real time functionality of LIVE-CHART,

it is important to allow teachers time to test the functionality and utility of the tool in real classroom settings. The initial version will allow students to select an option to raise their hands and articulate a question that is then sent to the teacher's LIVE-CHART display. Allowing the teacher to be able to address such student questions is vital to the implementation of the intervention. Subsequent development on additional functionality, such as allowing teachers to direct questions to other students, is also planned to be implemented by the end of this stage. The goal is to allow the Development Teachers the opportunity to use all aspects of the tool and provide feedback on the usage (in addition to other design elements) before evaluating the tool with the Pilot Teachers during the subsequent academic year.

Timeline for Participants

The timeline for the Development Teachers and Pilot teachers is illustrated by Figure 5 over the 3 year period of the grant. The timeline focuses early on the iterative development of the system using feedback from the Development Teachers as detailed in the previous section, while working toward the final pilot version of the intervention to be deployed to the Pilot Teachers in Fall of 2020. The Pilot Teachers will participate in several in-class live training/demonstration sessions that will occur at the beginning and end of the final two full academic years as will be detailed further in Section C.3. During the last academic year (2020-2021), the Pilot teachers will use LIVE-CHART and HAND-RAISE in their classrooms, allowing for final analyses and evaluation of the intervention during the final months of the grant period.

Figure 5: The timeline for participants.

8.3.3 Method of Evaluation

The evaluation of the HAND RAISE Intervention as delivered through the LIVE CHART tool focuses on three overarching research questions in alignment with the project's goals: improving students' confidence and skill in asking and articulating questions, improving students' confidence in answering questions of both the teacher and those of their peers, and helping to prevent the decline of interest in STEM-related fields that is observed between kindergarten and high school. We plan to evaluate the effectiveness of the HAND-RAISE intervention along these three dimensions through the use of surveys and quantitative field observations collected in Years 2 and 3 of the grant period. The next 3 sections detail this planned evaluation process.

Does the HAND-RAISE Intervention build student confidence to ask questions and diversify the students who do ask questions in the classroom?

It is important for students to feel comfortable in their learning environment to ask questions when they need more information, clarification, or further instruction to complete their work. It is similarly important for students to build the skill set of being able to effectively articulate their questions so that a teacher, instructor, or even peer can address the problem; it often difficult to help a student who simply says "I don't get it" when asked to articulate a question about the material. Conversely, however, we certainly do not want students to take advantage of the system as is sometimes exhibited in the over-use of hints; asking too many questions may be indicative of a student attempting to game the system, asking for help before applying themselves to learning the assigned topic. It is for these reasons that an integral aspect of HAND-RAISE is that it requires students to articulate their

question when they use the tool; certainly some students will still give the "I don't get it" response, but then the teacher is equipped with the ability to require a student to re-articulate the question before s/he addresses the problem. Similarly, the teacher has control over what the resulting action is when a student uses HAND-RAISE from the system; through LIVE-CHART, the teacher is able to use the reported information to determine if the student is using the tool effectively and respond accordingly by either speaking with the student, allowing another peer to answer the question, ask the student to re-articulate their question, or even instruct the student to try the problem before using the tool.

In order to evaluate the intervention on its effectiveness in improving student question asking behavior, we will use quantitative field observations collected by Dr. Kreisberg during a set of live in-class teacher trainings/demonstrations. Dr. Kreisberg will, as part of the intended teacher trainings, attend each of the Pilot Teachers' classes at the beginning and end of the academic years coinciding with years 2 and 3 of the grant. The teachers will be instructed beforehand to assign a selected homework assignment through ASSISTments for the preceding night, and Dr. Kreisberg will lead a homework review/discussion to demonstrate effective practices. Dr. Kreisberg will be equipped with her own version of LIVE-CHART for each classroom designed specifically for data collection. During this session, Dr. Kreisberg will present the students with a poorly-formed question pertaining to the content from their previous night's homework, in that a necessary piece of information will be omitted from the problem description; she will ask students to spend a few minutes to solve the problem, and turn her attention to her tablet so as to pretend not to see the hands that undoubtedly will begin to raise. Dr. Kreisberg will use her version of LIVE-CHART to record the students who have raised their hand and, after one minute to give students an opportunity to raise their hands,

will call on a student and take their question (likely pertaining to the missing piece of information).

In year 2, as the Pilot Teachers will not yet have used LIVE-CHART or HAND-RAISE in their classroom, the collected observations of students who raise their hands will act as a baseline measure of comparison for observations collected in year 3, but also will help indicate how hand raising behavior normally changes from the beginning to the end of the school year. Observations will, again, be collected at the beginning and end of each academic year to help control for differences in content difficulty (when comparing across years) and observe changes in individual student behavior (within student from the beginning to the end of each school year). The collected observations will be used to determine if 1) the number of hand raises in the classroom increase when students are faced with insufficiently-formed or confusing problems and 2) if the diversity of students who raise their hands increases as a result of the HAND-RAISE intervention.

Does the HAND-RAISE Intervention build student confidence to answer questions of their teacher and peers and diversify the students who raise their hands to answer questions in the classroom?

In addition to the ability to articulate questions, it is important for students to also build confidence and be able to articulate answers to questions asked by a teacher or a peer. It is important for students to feel comfortable in raising their hand to answer a question when the solution is known as it helps the teacher properly assess who understands the material as she is introducing new topics. The HAND-RAISE tool addresses and attempts to build a base of confidence and skill set focused on articulating answers through support of peer assistance. If the teacher chooses to connect a student who has their hand raised as indicated through LIVE-CHART

with another student (whether a starred student or otherwise), or if the teacher has enabled the tool to automatically choose another student, this provides another student the opportunity to address a peer's question anonymously. The act of helping another student on a topic where the helper has demonstrated understanding of the topic is aimed at building confidence in not only solving problems, but actively helping others to solve problems.

To evaluate the effectiveness of the HAND-RAISE Intervention in improving students' confidence and ability to articulate answers to questions, we will similarly utilize quantitative field observations collected by Dr. Kreisberg during a set of live in-class teacher trainings/demonstrations. As described in the previous section, Dr. Kreisberg will go to the classrooms of the 25 Pilot Teachers at the beginning and end of each academic year coinciding with years 2 and 3 of the grant. Again, she will lead a homework review/discussion based on a known assignment given to students as homework for the preceding night. From this assignment, Dr. Kreisberg will pre-select a problem to use as an example during the review session. Dr. Kreisberg's LIVE-CHART tool, specialized to help in the collection of observations, will display each student's performance on the pre-selected problem on the classroom display such that she is able to see which students answered the problem correctly; the problem will be pre-selected based on difficulty in an effort to maximize the number of students who answered the problem correctly. With this information, Dr. Kreisberg will display the problem to the class and ask which students can provide a solution, prompting students to raise their hands. Dr. Kreisberg will then pause to give students an opportunity to raise their hands to offer a solution and record such students through her LIVE-CHART tool before then calling on a student and proceeding. With the observations of which students raised their hands, a measure of effectiveness can be calculated as a percentage of students who knew the solution

(as indicated by which students answered the problem correctly on the homework assignment) raised their hand to offer an answer.

This measure, as compared from year 2 without use of the HAND-RAISE and LIVE-CHART tools to year 3 with such tools, and also from the beginning of each academic year to the end of each academic year, will give an indication of how student hand raising behavior for the purpose of answer questions is impacted by use of such tools in the classroom. Similarly, the described metric (percentage of students who know the answer that raise their hands) can be observed within smaller subgroups of students to understand if there are heterogeneous effects across students; specifically, as previous works have identified female and minority students as being less likely to engage in classroom discussions and activities with the same level of interest as some of their peers (Greenfield, 1997; Bernacki et al., 2016), it is a goal of this project to improve not only the help-seeking behavior of such students but also their level of engagement in answering questions during classroom discussion.

Does the HAND-RAISE Intervention help to reduce the decline of interest in STEM-related fields?

The decline of interest and motivation pertaining to STEM-related subjects from kindergarten through high school has been well-studied and documented in the United States (Alexander et al., 2012; Gottfried et al., 2009; Sorge, 2007; George, 2006) as well as internationally (Potvin & Hasni, 2014; OECD, 2006; Sjberg, & Schreiner, 2005; Osborne and Dillon, 2008). Several explanations have been offered to explain this decline as listed briefly in Section A, but it is likely that there is no single cause, suggesting that there is likely no single "one size fits all" solution. However, building better teacher-student and student-peer interactions, particularly

in cases of help-seeking and answering behaviors, is a promising area to focus in order to build confidence and increase student engagement in the classroom; through such confidence and engagement, more opportunities arise to build student interest in STEM. It is for this reason that the HAND-RAISE Intervention provides the necessary tools to allow the teacher to help facilitate these types of interactions while students are working in class and providing an environment aimed at supporting student engagement and confidence when asking and answering questions.

It is the aim of this intervention to increase student engagement, confidence, and interest in STEM-related subjects to reduce the widely-documented decline of such constructs over the course of the school year. In order to measure and evaluate the impact of the HAND-RAISE Intervention on student interest, particularly in that of math as it is this project's domain of focus, we will use a series of surveys given to the students of the 25 Pilot Teachers over the course of the academic years coinciding with years 2 and 3 of the grant. These surveys will help to gain a sense of each student's interest and perceived engagement toward math and STEM, their level of confidence in raising their hands in class to ask and answer questions, and also their sense of belonging amongst their peers in the classroom environment. We will derive the relevant survey items from previously developed, studied, and validated sources (Mahoney, 2010; Ostrow, 2018) and distribute the surveys to students through the Pilot Teachers at the beginning and end of year 2, before use of the tool in the classroom to measure the normal decline of these measures over a single school year, and then at the beginning and end of year 3 to measure how the use of HAND-RAISE in the classroom impacts each of these measures.

Similarly as is planned for the evaluation methods described in Sections C.3.a and C.3.b, we will explore potential heterogeneous effects within subgroups of students. Particularly, as larger declines of interest have been observed in female and minority

students, we will focus on such students to measure any potential effects of the tool.

8.4 Broader Impacts

The development and deployment of the HAND-RAISE Intervention to real classroom environments opens several opportunities to help develop better-informed teacher interactions and support the development of positive help-seeking and question-answer behaviors that have potential to expand beyond the use cases described in this proposal. The development of such skill sets are vital to success in STEM-related fields and can help foster better student achievement and engagement in this educational subjects. By focussing on student behaviors and performance as it occurs in the classroom, this project can take advantage of the opportunities made possible through the use of computer-based learning platforms to lead to positive impacts on student learning while helping prevent the decline of student interest in STEM fields.

8.5 Intellectual merit

The proposed project will help us better understand how teacher-student interactions and student help-seeking and question-answering behaviors impact engagement and interest in STEM-related subjects.

Part III

Understanding the Role of Student Knowledge, Behavior, and Affect in Productive Perseverance

Chapter 9

Studying Affect Dynamics and Chronometry Using Sensor-Free Detectors

Botelho, A. F., Baker, R. S., Ocumpaugh, J., & Heffernan, N. T. (2018, July). Studying Affect Dynamics and Chronometry Using Sensor-Free Detectors. In *Proceedings of the 11th International Conference on Educational Data Mining*, 157-166.

Abstract

Student affect has been found to correlate with short- and long-term learning outcomes, including college attendance as well as interest and involvement in Science, Technology, Engineering, and Mathematics (STEM) careers. However, there still remain significant questions about the processes by which affect shifts and develops during the learning process. Much of this research can be split into affect dynamics, the study of the temporal transitions between affective states, and affective chronometry, the study of how an affect state emerges and dissipates over time. Thus far, these affective processes have been

primarily studied using field observations, sensors, or student self-report measures; however, these approaches can be coarse, and obtaining finer-grained data produces challenges to data fidelity. Recent developments in sensor-free detectors of student affect, utilizing only the data from student interactions with a computer-based learning platform, open an opportunity to study affect dynamics and chronometry at moment-to-moment levels of granularity. This work presents a novel approach, applying sensor-free detectors to study these two prominent problems in affective research.

9.1 Introduction

The various affective states experienced by students during learning have received significant attention from the research community for their prominence in the learning process. Student affect has been shown to correlate with several measures of student achievement [CGSG04][PBSP+14][RBJ+09], has been found to be predictive of whether students attend college several years later [PBBH13], and also whether students choose to take steps towards careers in Science, Technology, Engineering, and Mathematics (STEM) fields [SPOBH14]. While significant steps have been taken toward understanding the inter-relationships between of affect and learning, there are many questions that remain unanswered with regard to how affect is exhibited by students over time as well has how such temporal trends may be informative of student learning outcomes.

The temporality of student affect has been characterized into two areas of study, affect dynamics [SC74] and affective chronometry. Affect dynamics studies temporal shifts in affect to understand which transitions between affective states are most common. A theoretically-grounded model of affective dynamics has been proposed by D'Mello and Graesser [DG12], which suggests a typical resolution cycle, where

students transition from engaged concentration to surprise to confusion and back to engaged concentration, but which also hypothesizes alternative transitions, including a path from confusion to frustration and boredom.

Affective chronometry also uses temporal measures, but focuses more closely upon how individual affective states (e.g., boredom) behave over time. This was first studied as a special case of affective dynamics, where researchers investigated how frequent it was for an affective state to transition to itself (aka "self-transitions"). More recently, D'Mello and Graesser [DG11] proposed instead investigating an affective state's "half life," or the decay in the probability of an affective state persisting for a specific duration of time. [DG11] found evidence that six affective states exhibit exponential decay in their probability over time. That is, the probability that a student remains in a particular state decreases exponentially as the amount of time that the student persists in that state increases. However, engaged concentration (referred to as flow) showed a much slower decay rate than other affective states (e.g., frustration).

There is now a growing body of research in affective dynamics and affective chronometry, commonly using field observations [RBA+11][GSR+11], or self-reports accompanied by video data [BD13][DG11]. These important studies have helped to advance the field, but each method imposes different kinds of limitations on the grain-size of the data. Continuous observation is impractical both for self-report and field observation studies, and it is highly time-consuming for video recording (which can also break down when the student moves away from his or her desk, either for off-task reasons or for on-task purposes like peer-tutoring or requesting assistance). Despite the limitations of these methods, they have often been preferred to sensor-free detectors of affect due to higher reliability/quality of the data obtained. However, recent advances in sensor-free detection of affect, based on deep

learning methods, have substantially increased the quality of models [BBH17], making interaction-based detectors a viable alternative. While these models are also not without limitations, their improved performance provides an alternative that facilitates near-continuous labeling at scale. As such, the recent advent of higher-quality detectors introduce the opportunity to study affect dynamics and affective chronometry with fine levels of granularity at scale.

In this paper, we present research studying affect dynamics and affective chronometry with the use of deep learning sensor-free affect detectors. We report the affect dynamics and chronometry for four commonly-studied affective states: engaged concentration [Csi90] (also referred to as engagement, flow, and equilibrium), boredom [Csi90][Mis96], confusion [CGSG04][KRP01], and frustration [KRP01][PSC93]. We investigate these relationships in the real-world learning of just under a thousand students, and compare our findings to prominent foundational research [DG11][DG12].

9.2 Previous Work

The theoretical model of affective dynamics proposed by D'Mello and Graesser [DG12] has become widely recognized in the study of affective state transitions. The model proposes a set of theoretically hypothesized transitions that have emerged through the study of student affect, as illustrated by the simplified representation of the model in Figure 9.1. While the full model observes numerous affective states including surprise and delight, we restrict the analysis in this paper to the key affective states of engaged concentration, boredom, confusion, and frustration.

The model hypothesizes that specific transitions between affective states are particularly common. In this model, a student commonly begins in a state of equilibrium (i.e. flow or engaged concentration). The student remains in this state until

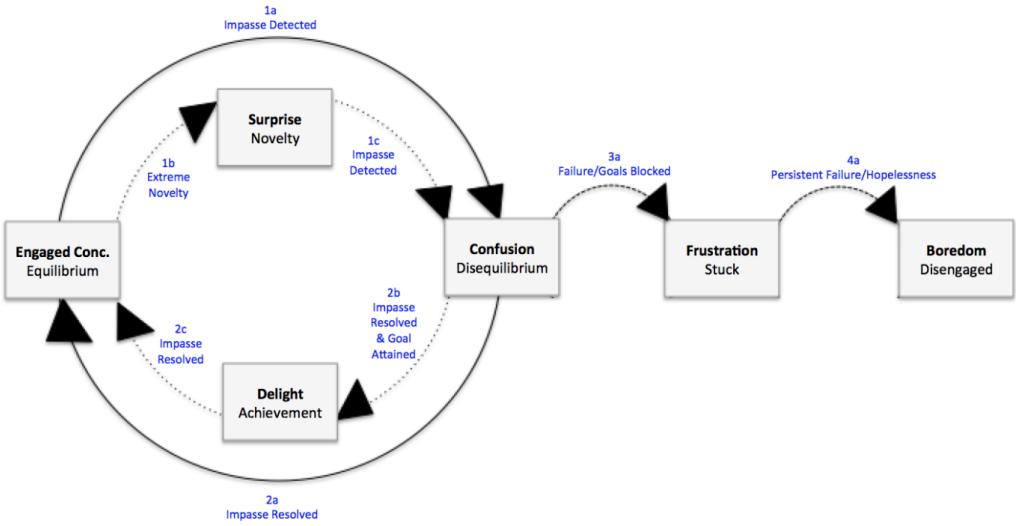

Figure 9.1: The proposed theoretical model of affect dynamics as presented by D'Mello and Graesser [DG12]

novelty or difficulty emerges, at which point the student may transition to confusion. The student may transition back to engaged concentration by resolving this confusion, possibly experiencing delight upon the way. Alternatively, the student my transition from confusion to frustration, at which point the model suggests that the student is unlikely to transition back to the more productive cycle of engaged concentration and confusion; instead, the student is more likely to transition from frustration to boredom. As such, while students may be expected to oscillate between certain adjacent states in the model, the model suggests that it is unlikely for students to transition to unconnected states as depicted in Figure 9.1.

The model has been explored in several studies [RBA+12][DG10] observing differences in student affect, and has become influential to other research studying affect dynamics in the context of other constructs such as gaming the system [RBA+11]. Other studies prior to the publication of this model also studied affective dynamics [BRX07][RRMB+08]. While the specific affective states studied across these projects

vary, the four affective states studied in this work are among the most commonly observed in this area of research. However, work in other paradigms also exists; for example, Redondo [Red16] attempted to identify when a student's affect shifts from increasingly positive to becoming more negative, or vice-versa, in self-report Likert scale data, finding that unexpectedly positive or negative affect typically indicated a shift in overall affective trajectory. However, she did not compare the prevalence of turning points found to overall base rates of affect, or analyze the chronometry of the sequences she studied. In general, across these papers, estimates of student affect have been collected through a range of methodologies including, most commonly, quantitative field observations (QFOs) [GSR+11][GRD+13][RBA+11][OAB+17], but also through self-reports in conjunction with post-hoc judgements of recorded video [BD13][BD17].

While there have been a large number of projects investigating affective dynamics, there has been substantially less research pertaining to affective chronometry. The study of affective chronometry is at times seen in affective dynamics papers. Among the papers investigating affective dynamics, several studies, including that of Baker, Rodrigo, and Xolocotzin [BRX07] have found that state self-transitions, where the student is in the same affective state in one observation as in the previous observation, were often statistically significantly more likely than chance. This suggests that students in each state do tend to persist for at least the duration of the time interval between observations (1 minute in that article); however, this paper did not observe the chronometry beyond this interval. In foundational work in this area, D'Mello and Graesser [DG11] investigated the duration of different affective states, proposing a methodology with which to evaluate the "half-life," or decay of individual affective states experienced by students. Using a computer-based system known as AutoTutor, the authors used a combination of self-reports of the students

and expert and peer judgments of student affect made using recorded video in order to measure and evaluate the length of time students commonly remained in each experienced affective state. However, that work was conducted on a relatively small number of subjects working on AutoTutor in a lab setting, on a task not related to their studies. It is therefore unclear whether the findings obtained in that context will generalize to data from a classroom environment where students are working on authentic educational tasks. The same methodology for measurement and evaluation of affective chronometry as presented in that work will be applied here to understand and compare affective chronometry – however, instead of using self-report, this project will utilize sensor-free detectors of affect applied to data collected from real students working in classroom environments.

9.2.1 Detectors of Student Affect

We apply the sensor-free detectors of student affect previously described in Botelho et al. [BBH17] to our data in order to study affective dynamics and chronometry. We use the same data set in this work from which the training set originally used in Botelho et al. [BBH17] was sampled, to ensure maximum validity of the detectors. In applying the detectors to this data set, we determined that several minor adjustments needed to be made to the detectors, so that the training data set was aligned to the ground truth observations in a way that could be more easily applied to the unlabeled data. We also reduced the number of features used as input to the model building algorithm. The detectors were refit using this adjusted dataset and produced performance metrics comparable to the previous work (average AUC = .74, average Cohen's Kappa = 0.20).

As in Botelho et al. [BBH17], these sensor-free detectors were developed using a long short term memory (LSTM) [HS97] network, a type of deep learning model

designed for time series data. LSTM networks use a large number of learned parameters with internal memory that can model temporal trends within the data to make estimates that are better informed by previous time steps within the series. Although the initial training sample was imbalanced, the use of resampling did not improve model performance, and a min-max estimate scaling was used instead. The LSTM model is trained as a sequence-to-sequence model, meaning that it accepts an entire sequence of time steps as input and produces a sequence of outputs. These outputs are in the form of a sequence of estimates of the probability that each of four affective states of engaged concentration, boredom, confusion, and frustration are occurring at each 20-second time step, or "clip," within the data. We use this sequence of probabilities to study affective dynamics and chronometry – the details of these analyses are provided in later sections. The LSTM model was found to produce cross-validated AUC values that substantially outperformed prior sensor-free detectors, which had previously exhibited an average AUC = 0.66, developed using older algorithms with the same dataset [OBG+14][WHH15]. In addition, LSTM models are designed to exploit the temporal character of the data, suggesting that they will be able to model temporal changes and transitions between affective state better than a model that treats each 20-second clip of student behavior as an independent sample.

9.3 Methodology

9.3.1 Dataset

The data[1] used in this work is comprised of action-level student data collected within the ASSISTments learning platform [HH14]. ASSISTments is a computer-

[1]The data used in this work is made available at http://tiny.cc/EDM2018_affectdata

based learning system used daily by thousands of students in real classrooms (over 50,000 a year) and hosts primarily middle school math content. The system has been used in several previous papers to study student affect, in many cases using sensor-free detectors of student affect.

Within this paper, we utilize a dataset originally used to develop sensor-free automated detectors of student affect. Detectors were originally developed using data collected by conducting field observations of student affect as 838 students used ASSISTments. 3,127 20-second field observations were collected in total, with gaps between one and several minutes between observations of the same student. For this paper, we analyze the entire data set of interaction for those 838 students on the days when observation occurred, 48,276 20-second segments of student behavior in total. We format the data in terms of 20-second segments of behavior in order to use the sensor-free detectors of affect, which were developed at this grain size (in line with the original field observations, which were conducted at the same grain size). The original training data set was highly imbalanced, with approximately 82% of observations coded as engaged concentration, 10% coded as boredom, 4% coded as confused, and 4% coded as frustration. This imbalance is consistent with previous research on the prevalence of these affective categories in systems such as ASSISTments.

The sensor-free LSTM detectors were applied to this dataset, providing an estimate of the probability of each of the four observed affective states for each of the 20-second segments of behavior within the system. The ground-truth labels used in model training are removed from this dataset and instead are replaced with the estimates produced by the sensor-free detectors. We replaced the ground-truth labels with the detector outputs so that the data would be comparable across all of the 48,276 observations.

9.3.2 Affect Dynamics

The estimates produced by the sensor-free detectors, when applied to the analysis dataset, are used to observe which transitions between affective states are frequent and statistically significantly more likely than chance. As is described in the previous section, the model produces four continuous-valued estimates corresponding with the 4 affective states of engaged concentration, boredom, confusion, and frustration. However, these estimates must be discretized and reduced to a single label describing the most likely affective state exhibited by the student at each time step. It is not sufficient to simply conclude that the most probable affective state (e.g. the affective state with the highest confidence) is the current affective state. For example, the model may predict very small values for all four affective states.

Instead, we first select a threshold that indicates that a specific affective state is likely occurring during a specific clip. We use a threshold of 0.5, defining a value above this threshold to be indicative of the presence of that corresponding affective state for the time step. 0.5 is a reasonable threshold as the detectors were previously run through a min-max scaling of the model outputs to remove majority class bias (cf. [BBH17]). However, there exists the possibility, as expressed in the example above, that no estimate across the four affective states surpasses this defined threshold. In such cases, a fifth "Neutral/Other" affective state is introduced to represent that none of the affective states we are studying is occurring; this state has been included in similar previous analyses of affect dynamics as well ([GSR+11][GRD+13][RRMB+08][RBA+12][BD17][DG11]). Conversely, it is possible for more than one estimate across the four outputs to surpass the defined threshold. In this unusual case (less than 1% of our data), no single affective state label can be applied and this clip (and transitions from and to this clip) is omitted from the subsequent analyses.

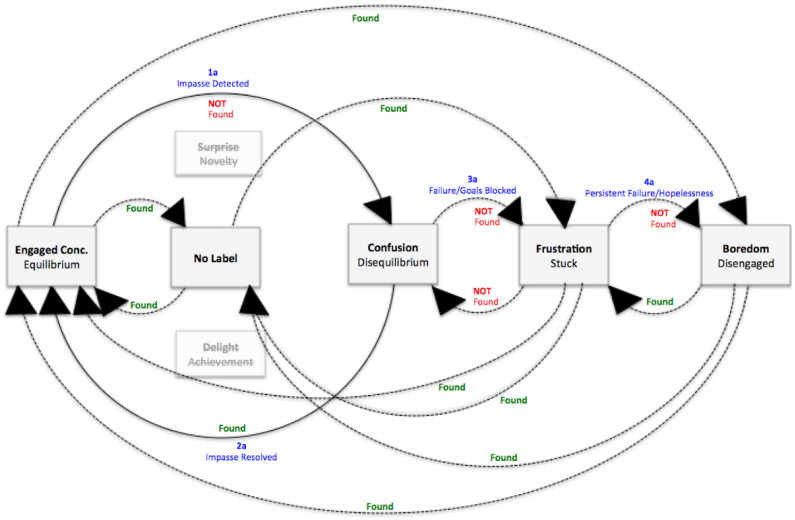

Figure 9.2: The resulting positive and significant affect transitions as compared to the D'Mello and Graesser [DG12] theoretical model.

Once all estimates have been classified as either a single affective state or the neutral state, transitions between these states within each student are computed. As in [DG12], we omit self-transitions where the student remains in their current affective state; these are instead represented through affective chronometry (see next section). We report D'Mello's L [DTG12] as a measure of the commonality of each possible transition from a source affective state to a destination affective state along with a corresponding p-value denoting the probability of this frequency of transition being obtained by chance. The D'Mello's L metric can be interpreted in a similar manner to Cohen's kappa, describing the degree to which each transition is more (or less) likely than would be expected according to the overall proportion of occurrence of the destination affective state across all cases. Values of D'Mello's L below zero are less likely than chance; values above zero represent the percent more likely than chance the finding is. In other words, a D'Mello's L of 0.4 represents a transition that occurs 40% more often than would be expected from the destination state's base rate. We compute statistical significance of these transitions using the method

originally proposed in [DTG12] – D'Mello's L is computed for each student and transition, and then the set of transitions is compared to 0 using a one-sample two-tailed t-test. Benjamini and Hochberg's [BH95] correction is used to control for the substantial number of statistical comparisons conducted.

9.3.3 Affective Chronometry

Our methodology for affective chronometry closely follows that of D'Mello and Graesser [DG11], with whom we compare our findings. In their analysis, the rate of decay was calculated as a probability of each state persisting over a 60-80 second window, using affect labels aggregated across multiple observation methods including the use of self-reports and both peer- and expert-observers. The probability that each affective state persisted (i.e. $\Pr(\mathrm{E}_t = \mathrm{E}_{t+20})$) was computed for 20 second intervals within that window.

The analysis in this paper uses the same discretized affect labels described in the previous section, transforming a sequence of sets of four probabilities to a single most-likely affective state per clip. The sequence of labels is broken into a set of episodes of each affective state, where an episode describes a series of non-transitioning affect that starts when the student transitions into the state and ends when the student transitions out of the state. A cumulative sum of time, in seconds, is calculated for each episode to measure how long each student remained in each affective state. With this value, a probability that a state will persist beyond a defined number of seconds can be calculated.

Due to the nature of our affect detection approach, persistence is estimated in 20 second intervals. At each interval, the probability that a student remains in each current affective state is calculated for durations up to 300 seconds, or 5 minutes. The resulting 16 probabilities (for durations of 0, 20, 40, ... , 300 seconds) can then

be used to compare the rates of decay across each of the observed affective states.

9.4 Results

9.4.1 Observing Affect Dynamics

The affective state transitions, measured by D'Mello's L, are reported in Table 9.1 with accompanying significance. Aside from those transitions that occur to/from the neutral/other state, the most common significant transition appears to occur between confusion and engaged concentration, followed by that of frustration to engaged concentration. Contrary to the theoretical model proposed by D'Mello and Graesser [DG12], significant transitions are found between engaged concentration and boredom as well as from boredom to engaged concentration. The findings suggest that students do not transition between these states through others as in the proposed theoretical model, but can occur directly.

It is further illustrated in the table that no state is found to transition to confusion more likely than chance, for which there are several possible explanations. Confusion was the least-frequently detected state as estimated by the sensor-free model (under 1.0% of the dataset). As such, it is likely that there simply were not enough instances of detected confusion in the data to produce significant results, possibly because the model had difficulty detecting confusion, contributing to an under-sampling of this state as estimated by the model.

These positive and significant transitions as identified by Table 9.1 are illustrated in Figure 9.2 for better comparison to the theoretical model depicted in Figure 9.1. Not only do the already-identified transitions become clearer, the number of transitions occurring to and from the neutral/other state, listed simply as "no label" in that figure, are also made prominent. As described in the generation of this fifth

state, this represents those estimates where no model estimates across the four affective states exceeded the defined threshold. It is important to note that this state may not be a single state at all, but rather comprehensively represents all other affective states exhibited by students that are not observed in the analysis. As such, it is difficult to make meaningful claims or draw significant conclusions regarding transitions occurring to or from this state.

The divergence of the emerging transitions and the theoretical model indicate that there are fewer oscillations that are detected by the machine-learned method. While not included in the theoretical model, D'Mello and Graesser propose in the same work [DG12] that oscillations can occur between all adjacent affective states within the graph under certain conditions, but that is certainly not the case as seen in Figure 9.2 gained from the empirical results of this work. This suggests that the learned model finds that students do not commonly transition back and forth between states such as confusion and frustration as often as hypothesized by the theoretical model, but no other such cases emerge.

9.4.2 Observing Affective Chronometry

The results of our affective chronometry analysis illustrate the length of time students commonly spend in each affective state before transitioning to either another observed state or the neutral/other state. The results of this analysis, depicted in Figure 9.3, show notable differences in affective half-life between affective states. Engaged concentration and boredom exhibit much more gradual declines as opposed to both confusion and frustration which both exhibit steep and rapid decay. Just as was done in the previous work of D'Mello and Graesser [DG11], the decay can be quantified by fitting an exponential function to each of the observed states. Again, as the neutral/other state may comprehensively represent multiple states that are

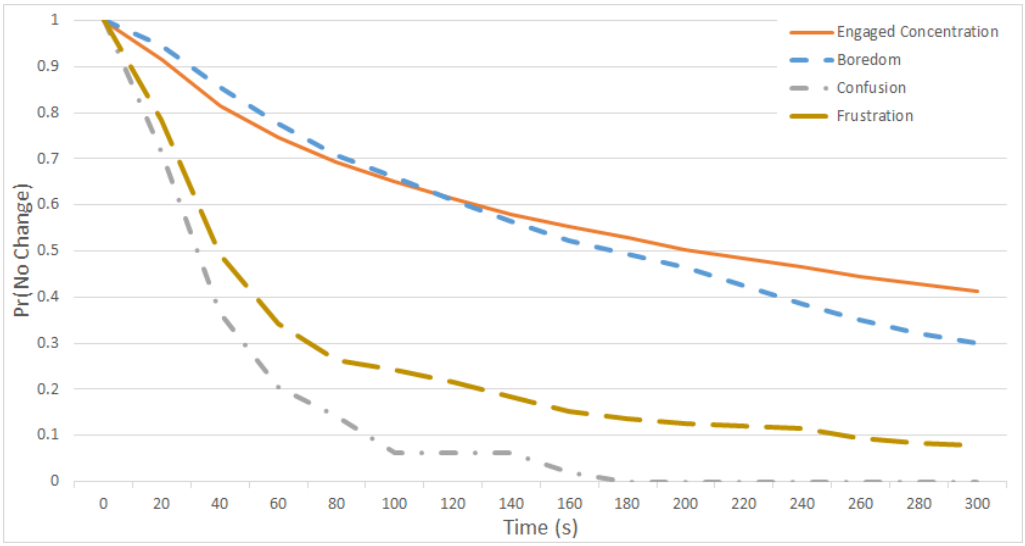

Figure 9.3: The probability of a student persisting in each affective state over time.

not measured in this work, this state is not included in the analyses of affective chronometry; if included, the results may simply illustrate an average decay over non-included affective states.

The value of decay for each state, as calculated by fitting an exponential curve to each states probability of persisting (Pr(No Change)) over time. Engaged concentration (decay = -0.003) and boredom (decay = -0.004) are found to have similarly gradual decay as compared to that of the remaining two states. Frustration (decay = -0.01) and confusion (decay = -0.024) are found to decay significantly faster. Of the studied states, only confusion is found to fail to persist past 5 minutes.

While the affective decay of engaged concentration, boredom, and frustration follow the general trend found by the work of D'Mello and Graesser in previous work [DG11], confusion deviates from this alignment. This difference is illustrated by Figures 9.4 and 9.5. Figure 9.4 illustrates the plotted exponential fit lines that were learned from the estimates produced by the sensor-free detectors. For comparison, Figure 9.5 illustrates the plotted exponential decay, as reported in Table 1 of D'Mello and Graesser [DG11]. From this, it becomes apparent that confusion is found to

exhibit similar decay patterns to that of engaged concentration and boredom, being more gradual over time, than that of frustration.

The other distinctive difference that emerges from the comparison of Figures 9.4 and 9.5 is that of the average time for decay across all affective states. This suggests that the average time that students remain in any affective state, as determined by the sensor-free model, is consistently longer than those found in D'Mello and Graesser [DG11]. The previous work reports that students rarely remained in a single state for longer than 60 seconds, and, following the learned exponential curve in Figure 9.5, no state seems to persist beyond 3 minutes, with most states reaching a probability of persisting close to 0 long before that time point. In comparison, each of the affective states, with the exception of confusion, are found to persist past the 5 minute time point, with engaged concentration and boredom seemingly persisting significantly beyond this point. Even in considering the 60 second timeframe, the fastest decaying state of confusion exhibits students persisting beyond this interval.

The divergence of the decay rates as exhibited by the estimates of the sensor-free model and those of the empirical findings reported in [DG11] may be due to a combination of differences between the two works. One possible explanation is the difference in learning contexts and the different learning interactions being studied in each of the two works. In this work, for example, the students comprising the dataset were in a classroom environment interacting with the computer-based system of ASSISTments. The previous study reported by [DG11], had students interacting with different software, namely that of AutoTutor, and also took place in a controlled lab setting. The domain of study also exhibits differences in that the students in AutoTutor were answering questions pertaining to computer literacy that are described as requiring students to answer in several sentences. The students using ASSISTments, however, were middle school students working on math content. The

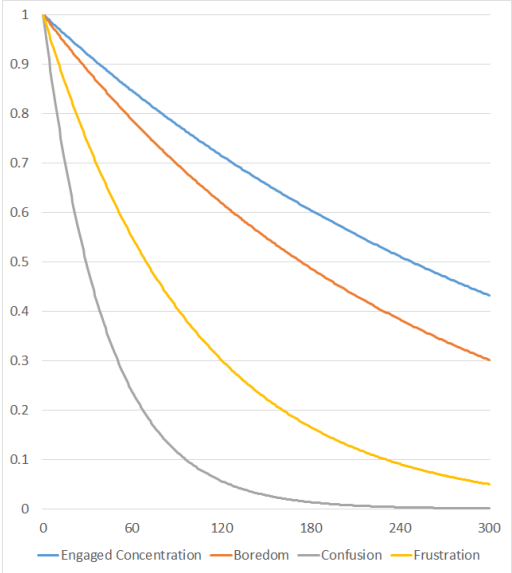

Figure 9.4: The plotted exponential decay of each affective state as estimated by the sensor-free affect detectors.

differences between both the content and the environment could have a distinct effect on the states of affect exhibited by students as well as the length of time students persist in each affective state.

9.5 Discussion and Future Work

The current work presents, to the knowledge of the authors, the first application of sensor-free affect detectors to study affect dynamics and affective chronometry. In studying affective dynamics, we can compare our results to a past theoretical model of affect dynamics proposed by D'Mello and Graesser [DG12], as well as other past empirical work. In affective chronometry, we can compare our results to past work [DG11], also by D'Mello and Graesser. The resulting model of affect dynamics produced by the application of sensor-free detectors shares little with the theorized model in regard to the significant transitions that emerged. Most notably, our model suggests oscillations between engaged concentration and boredom which

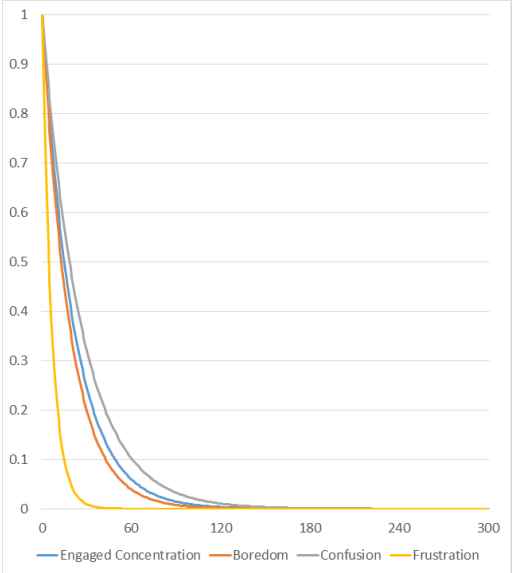

Figure 9.5: The plotted exponential decay of each affective state as reported in Table 1 of D'Mello and Graesser [DG11]

are hypothesized not to occur significantly in the theorized model; it has been found in other empirical work, however, that transitions between engaged concentration and boredom do appear [BD13][BD17]. The model of affective chronometry finds a similar pattern to D'Mello and Graesser in terms of which affective states are shorter and longer, but we find that all affective states last longer in our data set than in their previous work.

The application of sensor-free detectors to the study of student affect provides the opportunity to study how such affect is exhibited in students at greater scale and at second-by-second levels of granularity. In addition, automated detectors are a less intrusive method of data collection than more traditional methods. As the detectors utilize only data recorded from computer-based systems, they can estimate a student's affective state without interrupting their work, as can be the case with self-reporting methods, and does not hold a risk of observer effects where students change their behavior due to the presence of a human coder. The method also does

not require the use of additional technology such as physical and physiological sensors that may be difficult to deploy in classrooms at scale. Given the greater scale facilitated by automated affect detectors, future research may be able to study not just overall affective dynamics and chronometry but how dynamics and chronometry vary between different activities, different student populations, and even at different times of day. The better understanding of affective dynamics and chronometry that this may afford may have several benefits. Understanding a system's affective dynamics may be useful for encouraging positive transitions and suppressing negative transitions. Understanding affective chronometry may help us understand when negative emotion is problematic. Although some confusion is associated with positive learning outcomes [LDG12], extended confusion is associated with worse student performance [LPOB13]. Understanding whether a student's confusion or frustration lasts longer than the expected duration may indicate that a student is struggling and is in need of intervention.

As the scale of the application of automated detectors increases for the study of affective dynamics, the means of evaluating common transitions will likely need to evolve as well. After a certain data set size, all transitions will become significant. Even in this paper, with a relatively limited data set, fairly low values of D'Mello's L reached statistical significance. Future work may need to explore new methods of identifying and evaluating affect dynamics, perhaps by simply exploring reasonable means of leveraging D'Mello's L as a measure of magnitude to identify meaningfully frequent links, not just those that are simply statistically significantly more likely than chance.

There are potential limitations to the current work that may be addressed by future research in this area. First, while the sensor-free detectors used in this work, as presented in [BBH17], exhibit significantly superior performance to previous de-

veloped detectors with regard to AUC, improving the performance of these models further may help to improve transition and chronometry estimates, particularly of the less common labels of confusion and frustration. Utilizing methods to supplement less-frequently occurring labels of student affect (though the common method of resampling did not, in fact, enhance these detectors) or utilizing unlabeled data to better inform model estimates through co-training may improve model performance and produce more accurate measurements of affect dynamics and affective chronometry. It also may make sense to use different confidence thresholds for different affective states to adjust for the differences in the conservatism of different detectors that emerge from having different base rates.

Although consisting of a small portion of the data used in this work, the analyses did not include cases of co-occurring labels as estimated by the model. The estimates produced by the sensor-free detectors, even when the ground truth labels used to train such detectors did not observe co-occuring affective states themselves, is able to produce such cases, providing the opportunity to observe such cases in future work. Identifying which states are likely to co-occur, as well as include such cases in analyses of state transitions and affect state decay, will help to gain a better understanding of the relationships between affective states as well as to student performance.

A final opportunity for future work is in regard to observing affect dynamics and chronometry in experimental settings, as in the case of randomized controlled trials (RCTs). Several works have used analyses of state transitions to observe differences in affect exhibited between experimental conditions [RBA+12][DG10]. As the training set used to develop affect detectors does not contain experiment data, it is at this time uncertain if they generalize to behaviors exhibited outside of normal usage of the learning platform. Future work can observe how well such

detectors generalize to such populations of users and samples.

Table 9.1: The transitions between affective states. D'Mello's L values are shown. Transitions that are statistically significantly more likely than chance, after Benjamini and Hochberg's post-hoc correction, are denoted *.

From State	To State	D'Mello's L	p-value
Engaged Concentration	Engaged Concentration	—	—
	Boredom	0.260*	¡0.001
	Confusion	0.004	0.136
	Frustration	-0.12*	0.012
	Neutral/Other	0.481*	¡0.001
Boredom	Engaged Concentration	0.194*	¡0.001
	Boredom	—	—
	Confusion	-0.004	0.208
	Frustration	0.036*	¡0.001
	Neutral/Other	0.235*	¡0.001
Confusion	Engaged Concentration	0.341*	0.006
	Boredom	-0.127*	¡0.001
	Confusion	—	—
	Frustration	-0.026*	0.001
	Neutral/Other	-0.156	0.157
Frustration	Engaged Concentration	0.279*	¡0.001
	Boredom	-0.107*	¡0.001
	Confusion	0.008	0.391
	Frustration	—	—
	Neutral/Other	0.279*	¡0.001
Neutral/Other	Engaged Concentration	0.753*	¡0.001
	Boredom	-0.057*	¡0.001
	Confusion	0.003	0.302
	Frustration	0.015*	0.007
	Neutral/Other	—	—

Chapter 10

Refusing to Try: Characterizing Early Stopout on Student Assignments

Botelho, A. F., Varatharaj, A., VanInwegen, E., & Heffernan, N. T. (2019, March). Refusing to Try: Characterizing Early Stopout on Student Assignments. In *Proceedings of the 9th International Conference on Learning Analytics and Knowledge*, 391-400. ACM.

Abstract

A prominent issue faced by the education research community is that of student attrition. While large research efforts have been devoted to studying course-level attrition, widely referred to as dropout, less research has been focused on finer-grained assignment-level attrition commonly observed in K-12 classrooms. This later instantiation of attrition, referred to in this paper as "stopout," is characterized by students failing to complete their assigned work, but the cause of such behavior are not often known. This becomes a large

problem for educators and developers of learning platforms as students who give up on assignments early are missing opportunities to learn and practice the material which may affect future performance on related topics; similarly, it is difficult for researchers to develop, and subsequently difficult for computer-based systems to deploy interventions aimed at promoting productive persistence once a student has ceased interaction with the software. This difficulty highlights the importance to understand and identify early signs of stopout behavior in order to provide aid to students preemptively to promote productive persistence in their learning. While many cases of student stopout may be attributable to gaps in student knowledge and indicative of struggle, student attributes such as grit and persistence may be further affected by other factors. This work focuses on identifying different forms of stopout behavior in the context of middle school math by observing student behaviors at the sub-problem level. We find that students exhibit disproportionate stopout on the first problem of their assignments in comparison to stopout on subsequent problems, identifying a behavior that we call "refusal," and use the emerging patterns of student activity to better understand the potential causes underlying stopout behavior early in an assignment.

10.1 Introduction

Persistence is an essential factor of student learning as it is important for students to have the opportunity to work through problems and apply deliberate practice, particularly when exhibiting early struggle when learning new material. The study of this construct of learning has led to research into such student attributes as grit [DPMK07], perseverance [PS$^+$04], as well as other representations of high student persistence such as academic tenacity [DWC14], productive struggle [War15], and

productive failure [Kap08]. All of these theories of learning recognize that persistence is necessary in order for students to effectively overcome difficulties faced when learning new material. It is similarly understood that the lack of persistence can deprive students of the opportunity to effectively learn new and difficult material which may then propagate to affect the students' ability to learn subsequent post-requisite content. It is important, therefore, to ensure that students are able to take advantage of practice opportunities when they will be productive for learning and identify struggling students early to provide them with the help they need to succeed.

While not all representations of persistence are productive, such as the case of wheel spinning behavior (e.g. see [BG13]), it is often beneficial for students to exhibit high persistence during early learning opportunities. In this way, early student attrition becomes a significant problem for instructors and learning platforms as it is difficult to develop and deploy learning interventions and provide aid to students who cease interaction with the course or learning software. Not all student attrition, however, is exhibited in the same way and can emerge at varying levels of granularity.

With the emergence of massive open online courses (MOOCs), attrition in the form of student dropout has received a large amount of attention and research. The reasoning for which a student exhibits dropout, characterized as ceasing interaction with or explicitly leaving a course, has also been a well-studied problem within MOOCs [CRK15][XCSM16][YSAR13][RCY+14][LSHR15] as such courses often observe high attrition rates. Although dropout of this nature is not commonly observed in K-12 classrooms, attrition is still a prominent problem within this context and has received significantly less attention and research focus in previous years. Particularly as more classrooms begin to utilize computer-based learning platforms to assign classwork and homework, supplement instruction, and provide aid to students, there

are new opportunities to study student attrition at fine granular levels.

In the context of K-12 classrooms, it is common to observe student attrition at the assignment-level, where students begin an assignment but fail or choose not to complete the assigned work. This behavior, which we call "stopout," is distinctly different from the course-level dropout that is observed in MOOCs as students likely return to work on subsequent assignments; the student remains in the course, but did not finish the assigned work. Similar to the study of dropout, the reasoning for stopout behavior is not often known, but observing the immediate prior action that a student takes before stopout occurs within a given assignment may help to provide insight into the cause of the behavior. A student who exhibits stopout early in an assignment may do so for different reasons than a student who exhibits the behavior after attempting several problems, or learning opportunities as they will be referred in this work.

10.1.1 Student Refusal

In order to provide sufficient context for the goals and motivation of the current work, we must first describe a student behavior that emerged during a previous unpublished analysis of student stopout on a per-problem level conducted in 2015; this analysis is repeated here and will be described with greater detail in Section 4.2.

In observing when stopout occurs within student assignments, what quickly became apparent was that there seemed to be a disproportionate number of students exhibiting stopout on the first learning opportunity. Assuming that there would be a reasonably consistent failure rate over each opportunity, we found that student stopout by opportunity followed an exponential, or more specifically, Weibull distribution as is commonly observed in survival analyses [MM94]. However, while

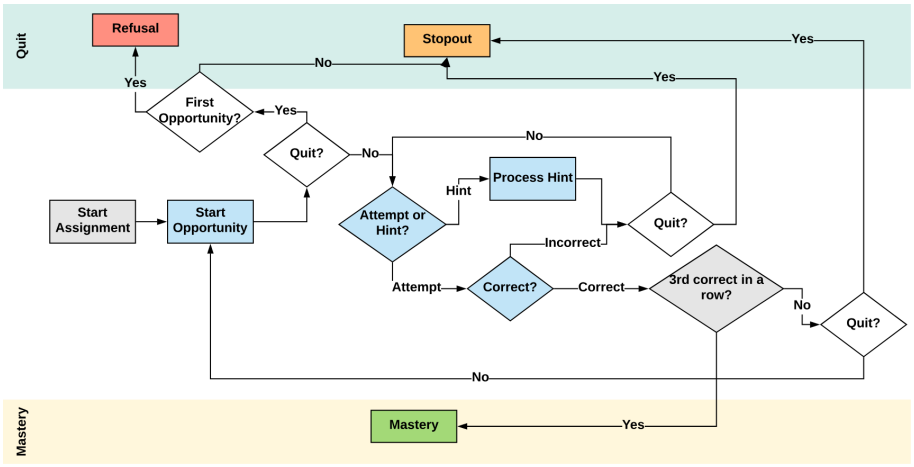

Figure 10.1: The flowchart of possible student actions resulting in either quitting (refusal or stopout) or mastery of the assignment.

most of the data followed this trend, the number of students exhibiting the behavior on the first opportunity was nearly double what would be expected by the fit exponential curve, as will also be demonstrated by Figure 10.4 in Section 4.2.

This behavior, which we call "refusal" was first used to identify problematic content within the learning system in which it was discovered, and is explored further in this work in an effort to better understand student interactions with the learning platform that may be indicative of early stopout behavior. The goal of this work is to explore the student actions associated with stopout and refusal behavior to better understand the potential causes of assignment-level student attrition within a computer-based learning platform. As students who exhibit refusal stop out of their assignments with little-to-no recorded interactions, it is these students who are arguably most important to identify in order to develop effective learning interventions to address any potential causes of this unproductive behavior.

In this research, we conduct a set of fine-grained analyses to determine the frequency of stopout as it correlates to the to the estimated knowledge level of each student in conjunction with the specific actions taken within the system immediately

prior to their stopout. We also then extend these analyses to include the dataset collected by Lang et al. [LHOW15] wherein they study the role of confidence on student learning using self-report surveys in a randomized controlled trial.

We seek to show in this paper that:

1. Student stopout after the first problem can be stochastically modelled as an exponential decay, but that this model fails to account for roughly half of the stopout that occurs on the first problem.

2. Specific actions (immediately prior to stopout) by students correlate with different patterns of stopout over time.

3. High stopout on the first problem correlates to low levels of self-reported confidence.

10.2 Background

The study of stopout in computer-based systems has largely focused on MOOCs in recognition of the often large attrition rates experienced by such courses. While the actions available to students in such courses often makes for feature-rich datasets with which to study attrition, the dropout behavior exhibited within such systems tend to observe contextual factors including the attitude of the student [CRK15], the estimated knowledge level of the student [KH15] combined with the effort exhibited by the student [YSAR13], as well as several other contextual factors such as technology, time management [WJ09], and other social factors [RCY+14].

Within these, however, it becomes clear that stopout behavior is not random but is seemingly motivated by more internal factors than external. The student is ultimately making the choice to dropout or stopout; many times, this is predictively

so [SS14], supporting the need to further understand why attrition occurs.

The problem of student stopout, however, is more prominent in K-12 classrooms than that of dropout experienced more in MOOC settings. In many cases, students choose to enroll in MOOCs, and can easily dropout due to a host of reasons briefly described above with little consequence. The problem of stopout in younger students is much more associated with a lack of persistence or motivation at an assignment-level rather than at the course-level.

The more general study of student persistence has led to a large amount of research exploring various aspects of the construct. Connotatively, persistence is often associated with positive learning behaviors, but in reality observes both beneficial and adverse effects depending on the context of which it is exhibited. It is intuitive that persistence can be beneficial when paired with productive learning behaviors, where learning occurs over time by making errors or receiving help. The productivity of persistence and perseverance is sometimes described by the construct of "grit" [DPMK07].

However, persistence may also be unproductive, as is the case of "wheel spinning" [BG13][GB15]. Wheel spinning describes the case when students attempt multiple problems but struggle to learn the material; this is analogous to a car that is stuck in mud or snow that "spins its wheels" but makes little to no progress. In such cases, stopout is sometimes encouraged as a more productive action, so long as the student takes such an opportunity to seek help from an instructor or parent.

In this work, we examine student behaviors that suggest a lack of persistence, i.e. when students stopout early in the assignment. While stopout may be encouraged in very select scenarios, as in the case of wheel spinning, it is generally considered a negative learning behavior as students lose the opportunity to learn through additional practice opportunities.

10.3 Dataset

The dataset used in this work consists of student log data collected as real students work in ASSISTments [HH14][RFMM16], a web-based learning platform aimed at supporting teachers and providing students with immediate correctness feedback on homework and classwork. The system hosts content across K-12 grade levels and even some college content, but is focused largely on middle school math content. Within the system, teachers can use the content provided by the system or create their own to assign to their students. The data used in this work is comprised mastery-based assignments, referred to as "skill builders" within the system. These skill builders usually give students isomorphic questions (generated from one or closely related templates) that have been previously generated, but randomly presented to the students; templates and questions are tightly associated in a single skill or sub-skill. Since the problems that student see are randomly selected from a large pool, we examine data not per problem, but rather per opportunity - i.e. the first problem a student sees is opportunity 1, the second is opportunity 2, etc.

Within the ASSISTments system, after opening a given problem, students can either submit an answer (and will receive instant correctness feedback), or they may use a help feature, such as requesting a hint. Hints (the most common type of help in this dataset) are usually written as some version of a complete worked out solution, often broken into pieces; the last hint (colloquially referred to as the bottom-out hint) gives the answer to the problem. If a student enters an incorrect answer (or requests a hint), they may then enter any number of attempts and use as many or as few of the hints as needed; the student must enter the correct answer before they are able to proceed to the next question. In order to successfully complete a Skill Builder, a student must enter the correct answer on the first attempt, using no help

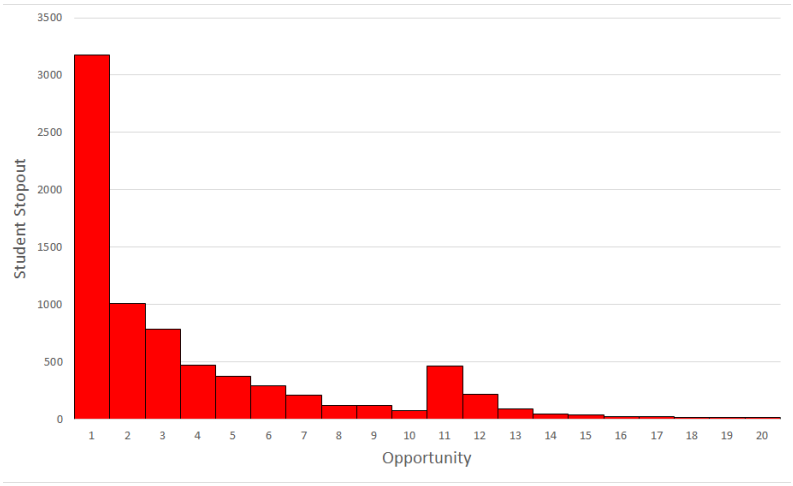

Figure 10.2: The frequency of student stopout by learning opportunity. Stopout on the first opportunity appears to be disproportionately larger than subsequent opportunities.

features, three times in a row.

Thus, at any given moment, a student can be said to be in one of three mutually exclusive conditions: Quit (either refusal or stopout), Working, or Mastery, as illustrated by Figure 10.1. The primary dataset in this analysis was taken from a previous school year; we also used the dataset from [LHOW15], which also comes from a prior academic year. Thus, when looking at the datasets, students have either attained mastery or have quit.

As we examine the behavior of students who have quit, we also note the action taken immediately prior to quitting. In ASSISTments, there are four possible actions a student may take before quitting a Skill Builder: they may have Opened an Opportunity (but have done nothing else), entered a Correct Attempt, entered an Incorrect Attempt, or made a Help Request. In this analysis, we make no differentiation of whether the help requested gave an initial step in the solution or the final answer.

In this paper, we will use the term stopout to refer to any student who leaves

(and never returns to) an unfinished assignment. Furthermore, for reasons discussed below, we refer to one specific type of stopout as refusal - that is, students who quit an assignment having only opened the first problem, without using any hint features or entering an attempt to answer it.

The data used in this work uses data from the 2016-2017 academic year and includes information recorded from 3,641 distinct students who exhibited stopout on skill builder assignments. Each row of the dataset corresponds to a single assignment attempted by a student. As this work is studying only those who exhibited stopout, students who complete each assignment are not included in the data or analyses. In an effort to remove cases where the completion of an assignment may have been optional, only assignments that had been started by at least 10 students and have an overall completion rate higher than 75% were considered for the analyses.

A second dataset, described further in Section 4.4, was also used to observed the relationship between stopout behavior and student confidence. This data consists of students interacting with the ASSISTments learning platform for a randomized controlled trial studying student confidence [LHOW15]. From the dataset used in that work, we extracted all students from the treatment condition (e.g. the students who received a confidence survey prior to beginning their assignment) who exhibited stopout during the assignment; this excludes any student who stopped out on the initial survey as well as students who finished the survey but did not begin the first non-survey problem of the assignment. The resulting dataset used in this work consists of 438 distinct students who exhibited stopout.

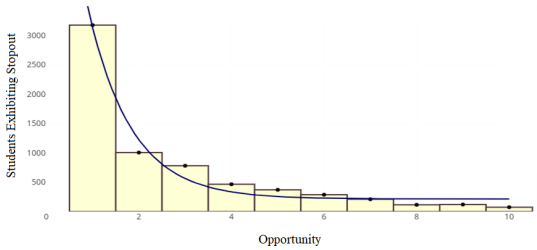

Figure 10.3: The exponential curve fit to stopout on the first ten learning opportunities. The line is a poor fit seemingly due to stopout on the first item.

10.4 Methodology

10.4.1 Characterizing Early Stopout and Refusal

It is important to clarify, before describing our analyses, how we have defined stopout within the data. In any sense, just as it has been described in earlier sections, stopout is exhibited when a student begins an assignment and fails or refuses to finish that assignment. It follows, then, that students who never begin an assignment did not exhibit stopout and are therefore not included in our data or analyses[1]. It is found that when students do stopout, however, it occurs after four distinct kinds of actions taken in the system. Students stopout either during a problem, or exhibit stopout after completing a problem but before progressing to the subsequent problem; in this later case, the student managed to enter the correct answer, but stopped out before seeing the next problem. In such a case, we mark the student as stopping out on the following opportunity. For example, if the student enters the correct answer to the first problem, or opportunity, but does not begin the second problem, that student is said to have stopped out on the second opportunity as the first problem was

[1]Although we would have preferred to include these students in our analyses, given the variety of grading policies of individual teachers we would be unable to determine how many students were required to complete an assignment, but never even opened it. We can state for certain how many students opened the assignment and failed to complete it; we cannot state for certain how many students should have opened the assignment, but did not.

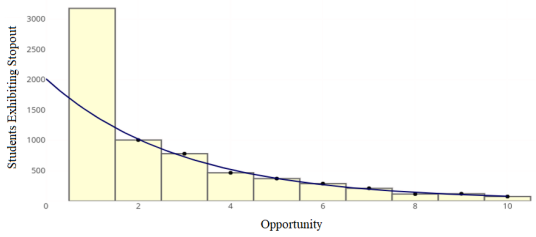

Figure 10.4: The exponential curve fit to stopout on opportunities 2 through 10, extended to show predicted stopout on the first problem.

sufficiently completed. When students stopout during a problem, before entering the correct response, those students are said to have stopped out on that learning opportunity (e.g. the student opens the first problem makes an incorrect attempt, or even no attempt, and then stops out is defined as the student stopping out on the first opportunity).

In order to better understand the behavior associated with stopout on skill builder assignments, it is important to first understand how stopout is exhibited independent of students and assignments. As was introduced in Section 1.1, we can explore this by simply observing the trends of stopout over all student assignments in the data. We first observe the distribution of where stopout occurs in an assignment by plotting the frequency of stopout by opportunity, as illustrated in Figure 10.2. Again, as introduced in Section 1.1, it is clear that there is a large number of students who stopout on the first, and subsequently the the eleventh opportunities; this observed spike on the eleventh opportunity can be attributed to students reaching the "daily limit" within the system which stops students who have not completed the assignment by the tenth opportunity, suggesting that they seek help and return to complete the subsequent day (e.g. to help prevent wheel spinning behavior). While the increased stopout observed on the eleventh opportunity to students who do not return after reaching the daily limit, no such reasoning can easily be given to explain the increased stopout observed on the first opportunity.

While visually it appears that there is disproportionate stopout on the first item as compared to subsequent opportunities, we first attempt to show this by exploring the modeling of stopout by opportunity. As the distribution appears to fit an exponential decay function, we fit two such curves to compare the goodness of model fit. We first fit an exponential curve to opportunities 1 through 10, as seen in Figure 10.3. We compare this model to another exponential curve that uses just opportunities 2 through 10, as seen in Figure 10.4. The comparison of these two models shows that there is disproportionate stopout that occurs on the first item. The R-squared values confirm this, with the first model exhibiting an R-squared value of .816 calculated over opportunities 2 through 10, and the second model exhibiting an R-squared value of .991 calculated over the same range. The model using just opportunities 2 through 10 fit an exponential curve nearly perfectly to the real data, illustrating where the expected stopout on the first opportunity is if it were to follow the same trend; in this regard, over twice as many students stopout on the first item as expected (an estimated 1,371 as compared to the observed 3,076 students). The observed difference between the expected and the observed number of students exhibiting stopout on the first learning opportunity is hypothesized to describe the estimated number of students exhibiting *refusal* as introduced in Section 1.1.

It is for this reason that it becomes even more pertinent to understand what causes so many students to exhibit refusal, as they stopout before even trying to learn the material. From this alone, it is unclear if students are exhibiting refusal due to a lack of knowledge or confidence, or if other behaviors are the cause, such as those associated with frustration or boredom. The analyses described in the next section, while non-causal, will help to provide insight into the behaviors associated with student stopout.

10.4.2 Categorizing Stopout Behavior

While the previous analysis observed stopout across all students, we further explore the behaviors associated with stopout for each student assignment. As described, there are several student level factors that may affect how the behavior is interpreted. For example, an estimated higher knowledge student who stops out on the first item without taking any action is likely to do so for different reasons than an estimated lower knowledge student with the same recorded activity; in the first sense, it may be boredom that causes the student to stop out after determining he/she is already comfortable with the material, while the later student may stopout due to low confidence in their ability to solve. It is likely that students cannot be dichotomized so cleanly, where a higher knowledge student stops out due to low confidence, but the analysis presented here will act as an initial step toward identifying these potential causes.

We use one student-level and 4 action-level covariates to group students by their last recorded activity before exhibiting stopout for each assignment. As the same student may stopout on different assignments for varying reasons, each student-assignment is treated as a separate sample, with grouping performed at the assignment level.

At the student-level, we estimate student knowledge based on the percent of correctly answered items attempted before beginning the observed assignment. This estimate will help to identify students who commonly answer problems correctly from those who often struggle to learn new material. As this covariate exhibits a positive skew, the value is squared to produce a more normal distribution representing estimated student knowledge. This transformed prior percent correct for each student will be used in subsequent analyses and referred to simply as prior correctness for simplicity.

The action-level covariates used in this work describe the last action recorded by the system for each student in each assignment. As all students in the dataset exhibited stopout, this represents the last activity taken by the student before stopping out of the assignment. Each action is represented as a binary value, and is limited to just the last action taken by the student. These actions are as follows:

- Opened Problem - denoting that the student opened the problem but made no subsequent action.
- Correct Attempt - the student entered a correct response to complete the problem, but did not progress to the subsequent problem.
- Help Request - the student requested an on-demand hint or scaffolded question, but made no further attempt to answer the problem.
- Incorrect Attempt - the student entered a response but the answer was incorrect.

We group students by their prior correctness and last recorded action using k-means clustering to gain an understanding of the different behaviors that emerge associated with student stopout. Determining the correct value of k in this type of analysis is important to the interpretability of the results. We determine this value using a short grid-search using different values of k between 2 and 15 and observing the variance of within-sum of squares between the emerging clusters similar to a skree plot used in principal component analysis. From this step, a value of 6 is determined to best partition the data; values 5 and 7 were additionally explored, but did not lead to large differences in interpretation, further supporting the usage of 6 groups to summarize the data.

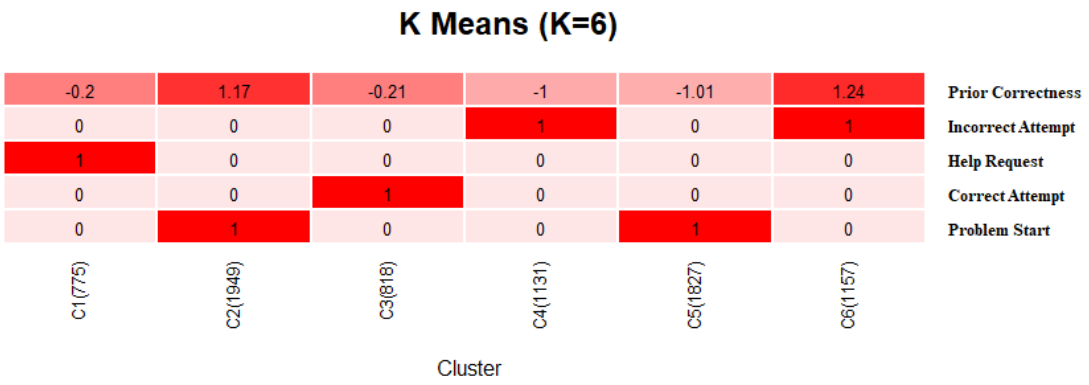

Figure 10.5: The resulting clusters of student prior correctness and last action pertaining to student stopout.

10.4.3 Stopout Behavior by Opportunity

Once student assignments have been grouped into the 6 clusters described in the previous section, we can further identify how the behaviors associated with stopout change with the opportunity. As we observe differential dropout on the first learning opportunity as compared with subsequent opportunities, we are hoping to observe differences in behaviors across learning opportunities to help explain this phenomenon. By observing how the distribution of the clusters changes with each learning opportunity, we can gain an understanding of which behaviors, if any, occur most on the first opportunity as compared to subsequent opportunities.

We limit our analysis to just the first three learning opportunities. As the number of students present decreases with each opportunity due to stopout, the number of students on later opportunities makes it difficult to make fair comparisons to earlier problems that are better represented by higher numbers of students. Additionally, as students know the threshold of completion being three consecutive correct responses, observing the first three opportunities highlights those students who exhibit the lowest persistence, stopping out on or before the earliest problem of which the assignment can be completed.

The distribution of the clusters is observed, filtering to include those who stopout on the first, second, and third opportunities and visualizing how this distribution changes. As fewer students are available for each opportunity, a proportional distribution is used by dividing the number of students included in each cluster by the total number of students who exhibit stopout at each respective opportunity.

10.4.4 Observing Student Confidence

Just as is the case with stopout behavior as a whole, refusal likely occurs as a result of many factors. In this work, however, we focus on exploring the relationship between two such possible factors with refusal behavior: lack of knowledge and confidence. As detailed in the description of our cluster analysis, we use prior correctness as an indicator of how well the student is expected to know the material; students who perform well on prior material often exhibit comparatively high performance on subsequent content as the student has demonstrated knowledge of foundational material. In this way, estimated knowledge, or lack thereof, can be explored amongst students exhibiting stopout and refusal behaviors.

In order to observe the relationship between these behaviors and confidence, however, we utilize an auxiliary dataset consisting of students who participated in a randomized controlled trial with the ASSISTments platform in an earlier academic year [LHOW15]. In this study, students assigned to the experimental condition were asked to answer a survey item before starting the assignment (and then subsequently asked again during the assignment, although only the initial survey was used in this work). Students were shown an example of the problems that would be seen in the assignment and asked them to self-report their level of confidence on a 5-point scale ranging from 0% (not confident at all) to 100% (very confident). Using the subsequent student data collected from the student assignments, we apply the

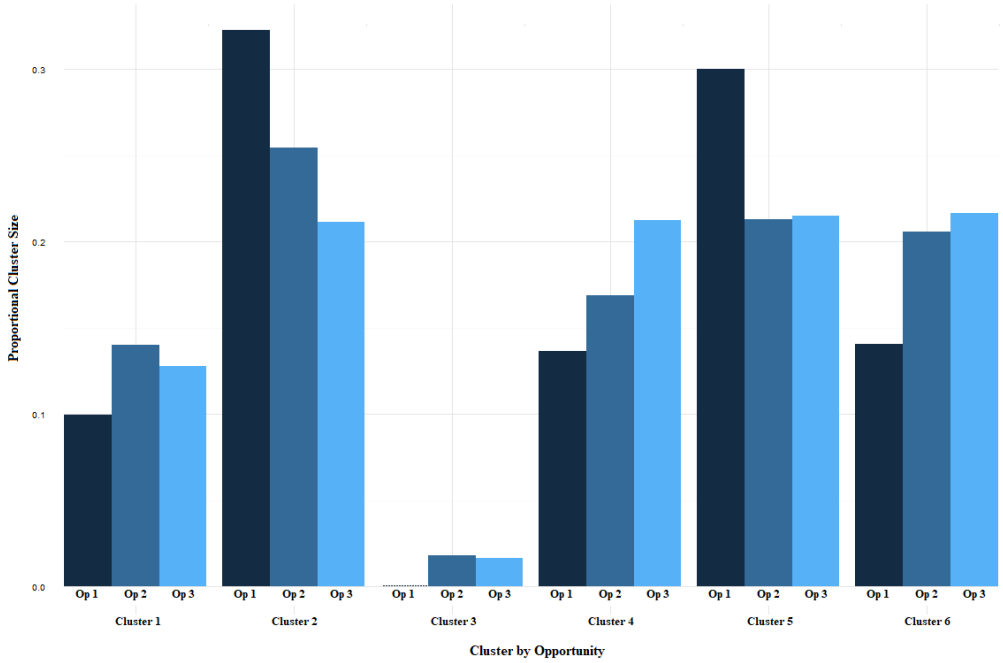

Figure 10.6: The proportional distribution of samples within each cluster over the first three learning opportunities.

clusters developed in Section 4.2 to observe any significant differences in reported confidence between each of the clusters. In regard to refusal behavior specifically, we also compare differences in reported confidence for students who exhibit stopout on the first opportunity.

10.5 Results and Discussion

The resulting 6 clusters of student prior knowledge and last recorded action is illustrated in Figure 10.5. Being the only continuous variable, the prior correctness appears to be a distinguishing factor among the student activity. This measure, being close to normally distributed after the described transformation, is represented as a z-scored value across the 6 groups in the figure; cluster 6, for example, represents the highest knowledge students who stopped out after an incorrect answer. Again,

this figure is the clustering as performed over the entire dataset independent of the learning opportunity on which students exhibited stopout. The resulting clusters further distinguish themselves by the last action taken by each student, with no cluster found to contain more than one type of action taken by students. This finding supports the claim that the stopout behavior is contextual, as it is not the case that a cluster represents, for example, estimated low knowledge students regardless of the last action taken.

The number of student assignments that fall within each cluster is denoted under each column along with the cluster number. From this, it becomes clear that the majority of students, regardless of high or low knowledge, stop out at the start of a problem without taking action as illustrated by clusters 2 and 5. The clusters with the fewest students, clusters 1 and 3, appear to have the lowest knowledge students who stop out after a help request and after a correct response respectively. The remaining groups, clusters 4 and 6, both contain students who exhibit stopout after an incorrect response, but represent opposing knowledge estimates.

While the clusters themselves seem to offer some interpretation as to the types of behaviors exhibited by students in the context of estimated knowledge, the final analysis offers an opportunity to observe these groupings by opportunity as well. Figure 10.6 depicts the results of this comparison, observing the distribution of student assignments that belong to each cluster by opportunity. Cluster 3 is found to have the fewest overall students proportionally in the first three opportunities; as this is not the smallest cluster when observing all student assignments, this suggests that this behavior is exhibited more on later opportunities. It is also the case, due to our definition of stopout, that no student can stopout on the first opportunity following a correct response. Aside from this, cluster 1 similarly contains the fewest number of students that also appears to be less affected by opportunity as no clear

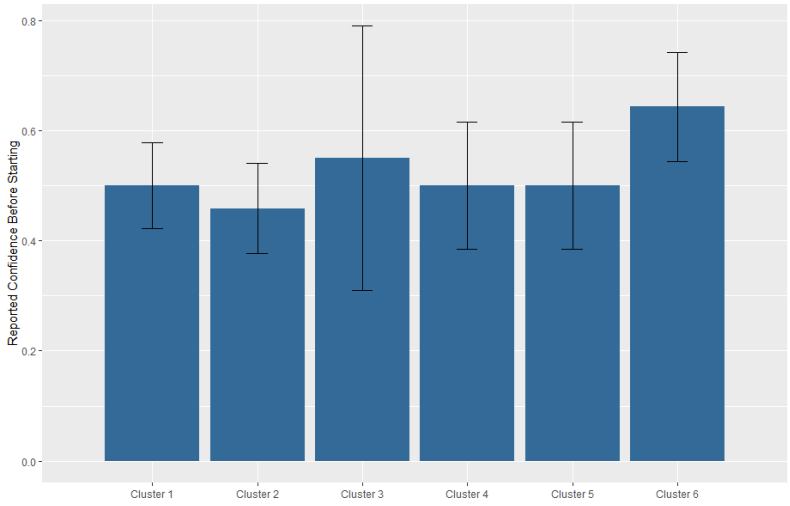

Figure 10.7: The reported confidence of students within each cluster with associated 95% confidence intervals.

trend emerges within this cluster.

The remaining four clusters, however, do exhibit interesting trends over the first three opportunities. Clusters 4 and 6 exhibit increasing numbers of students stopping out following incorrect responses, though distinguishable by the estimated knowledge level of students found within these clusters. Cluster 2 conversely exhibits a decreasing number of high knowledge students exhibiting stopout at the start of a problem before taking any further action. Finally, cluster 5 contains a notable trend in that the number of low knowledge students stopping out on the first item before taking action is noticeably higher than subsequent opportunities and exhibits no increasing or decreasing trend beyond this point within the observed opportunities. For this reason, it is likely that the cause for the disproportionate stopout on the first learning opportunity is largely due to students within clusters 2 and 5; these, again, are the students exhibiting refusal by our definition. Furthermore, the number of students who fall within clusters 2 and 5 on the first learning opportunity are 1,025 and 954, respectively, which, when subtracted from the total number of 3,076

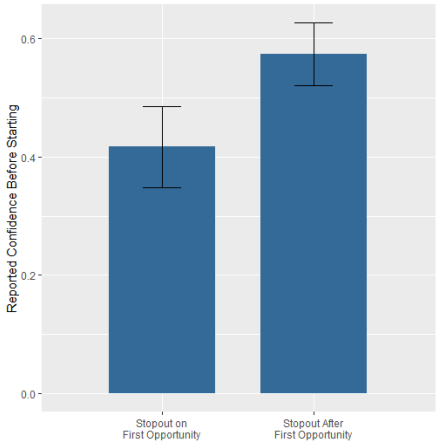

Figure 10.8: The reported confidence of students who stopout on the first learning opportunity as compared with students who stopout after the first learning opportunity with associated 95% confidence intervals.

students who exhibited stopout on the first opportunity as illustrated in Figure 10.2, the resulting 1,097 falls much closer to the expected 1,371 students as determined by our fit exponential model described in Section 4.1. We are not attempting to claim, of course, that this simple comparison of sample sizes fully explains the observed disproportionate stopout exhibited on the first learning opportunity, but the results of our analyses coupled with these comparisons do suggest that refusal behavior accounts for a majority of the phenomenon.

It is found, comparing the results of both the clustering analysis and comparison of cluster distributions across learning opportunities, that the disproportionate stopout tends to occur regardless of knowledge level, at the beginning of the problem before taking any action. This problem becomes more perplexing considering the effort to remove optional assignments using a completion threshold during data collection and filtering. Assuming that at least a majority of optional assignments and outlier cases are removed during that cleaning process, the fact that the two largest clusters are still comprised of those students who stopout without taking action further stresses the need to understand the definitive causes of such behavior.

The results of our final analyses are depicted in Figures 10.7 and 10.8, comparing the reported confidence measures of students by both cluster (Figure 10.7) and first opportunity versus subsequent opportunities (Figure 10.8). As the number of students who exhibited stopout in this supplementary dataset is significantly less than that observed in our earlier analyses, the 95% confidence intervals vary greatly. In observing Figure 10.7, for example, the majority of intervals overlap making us unable to claim reliable differences between many of the clusters. However, two clusters, 2 and 6, do emerge as significantly different with regard to the level of reported confidence. These two clusters represent the highest performing students compared to other clusters and yet exhibit vastly different levels of confidence, with the lower confident students being those who stopout without making any action in the problem. It is important to clarify that this figure includes students who stopout across all opportunities and not specifically those who stopout on the first opportunity (e.g. Cluster 2 here is not specifically students exhibiting refusal). It is also important to recognize that all reports of student confidence are reliably smaller than 0.8 (and several being even lower), suggesting that a large number of students who exhibited stopout, unsurprisingly, were not confident in their ability to successfully complete the assignment.

Figure 10.8 illustrates a significant difference found between the reported confidence of students who exhibit stopout on the first opportunity as compared to students who stopout on subsequent opportunities. It is important to clarify, however, that this comparison includes all students who stopout on the first opportunity in a single group as opposed to comparing students specifically exhibiting refusal (i.e. stopping on the first opportunity after taking no action) as it was found that very few students exhibited refusal in the supplementary dataset (only 4 students were found). This is contrary to the proportion that was found in other skill builder

dataset, but may be attributable to the context of the study; we believe refusal may occur as students realize that they are not confident in their ability to successfully complete the assignment, and as their confidence is revealed by the survey item, it is likely that students who would have exhibited refusal simply never began the assignment and subsequently would not exist in our dataset (as they saw no learning opportunities of the assignment). Despite this, we still see a significant difference between students who stopout on the first opportunity when compared to stopout on subsequent opportunities, suggesting that confidence, perhaps even more so than knowledge (in considering clusters 2 and 6 in Figure 10.5), is associated with refusal and early stopout behavior in student assignments.

10.6 Contributions and Future Work

The current work represents an initial step toward better understanding the causes of student stopout in K-12 classrooms by exploring the student actions and attributes associated with such behavior. With this in mind, this work can act as a foundation for future research aimed at finding more causal links between behavior and stopout as. A simple approach, as the students do not drop out of the respective courses, would be to survey students to determine the reasons for stopping out of an assignment.

There are several limitations to the current work that can be addressed with further research as well. The first is in the scope of the behaviors considered for grouping student assignments. In the analyses presented in this work, only the last action taken by the student was considered within the clustering. This feature can be vastly improved by generating more descriptive features of student activity or even by utilizing earlier information pertaining to each student. Another limitation of the

current work is the lack of contextual information pertaining to each assignment. The clustering is performed observing only student attributes as it is believed that this is most important to understand the behaviors associated with stopout, but understanding how these attributes interact with assignment-level features, such as the difficulty of the subject matter, may be helpful to understanding the concept as well.

Another limitation of the current work is the lack of causality of our analyses. While it is among the goals of this work to identify potential causes of stopout and refusal behavior, all analyses conducted are limited to correlation rather than causal claims. Future work may be able to address this by conducting randomized controlled trials aimed at identifying and deploying interventions to prevent potential stopout and refusal behaviors.

The contributions of the current work are 3-fold toward understanding the behaviors and actions associated with student assignment-level attrition in K-12 classrooms. First, the current work identified a disproportionate stopout on the first opportunity as compared with subsequent opportunities. While stopout tends to follow an exponential decay, this does not extend to the first learning opportunity. This highlights a need to research this phenomenon further to direct the development of learning interventions aimed at deterring students from giving up to early or too easily when faced with difficult content. We show in this work that a large proportion of this early stopout is likely attributable to a behavior we have identified as refusal.

The second contribution is in the exploration of student actions associated with stopout. With the 6 groups of student knowledge-action interactions that emerged from the analysis, these clusters form the basis to conduct further research exploring their predictive power in other aspects of student learning. These groups of students

highlight that low persistence, as defined by student stopout, is not exhibited in the same way across all students or even across students of similar prior knowledge. Furthermore, the actions associated with stopout behavior are found to change over each learning opportunity, suggesting that, unsurprisingly, the reason for stopout is dependent on where the behavior occurs within each assignment.

Finally, it is clear from this work, as well as the work of Lang et al. [LHOW15], that confidence is strongly related to student assignment-level attrition, perhaps even more so than gaps in student knowledge, supporting the need for learning interventions to address this factor to promote more productive learning practices. This confidence level, while comparatively low for all students who exhibited stopout in our analyses, appeared lowest for students who exhibited stopout behavior on the first learning opportunity. Similarly, the level of confidence for high knowledge students was divided between two of the identified clusters of students, suggesting that confidence is not directly dependent on prior knowledge.

Chapter 11

Identifying the Constructs Underlying Models of Student Knowledge, Behavior, and Affect

Botelho, A.F. (2019). *Identifying the Constructs Underlying Models of Student Knowledge, Behavior, and Affect.* Manuscript in Preparation

11.1 Introduction

Failure is a difficult yet inevitable aspect of the learning process, and a person's reaction to failure can impact later performance. This work represents, to the authors knowledge, the first such set of analyses aimed to look across a wide range of machine learning models developed to measure student knowledge, behavior, and affect in order to identify and explore the underlying represented learning constructs. In this way, this work seeks to bridge the gaps that exist between theory and methods and further validate and explore the deeper relationships between the constructs of

learning that are being measured.

This work explores these various detectors, sorted into a folksonomy consisting of the three categories of student knowledge, behavior, and affect, to explore the dimensionality of constructs measured. As the chosen detectors attempt to model different aspects of student engagement, it is hypothesized that the constructs that emerge will represent those theorized to be closely related to measures of productive and unproductive persistence. In this way, it is the primary goal of this work to bridge the gap in research that has been conducted on student engagement through the development and application of various detectors and observe how these measures relate to each other as well as distinguish productive and unproductive perseverance and how predictive these are of longer-term learning outcomes.

11.1.1 Given a variety of commonly used assessment measures of student success, what is the dimensionality of the constructs measured by these assessments?

Based on the prior research in education and learning analytics, it is hypothesized here that many commonly-observed student assessment measures are correlated. In other words, it has previously been observed that high performing students tend to consistently perform well while low performing students tend to perform poorly across assessments (c.f. [BWH15]). It is not clear, however, what the dimensionality of these measures are in regard to the constructs that are being measured.

11.1.2 What is the dimensionality of constructs measured by the observed detectors of student knowledge, behavior and affect?

Many of the detectors of student knowledge, behavior, and affect identified and described in the next section attempt to measure varying aspects of student engagement while working through a learning task. While many have been developed with different learning theories in mind (i.e. measuring behavioral constructs rather than affective), it is not unreasonable to assume that these detectors exhibit overlap in regard to the underlying constructs being measured. It is uncertain, however, the degree of overlap across these differing detectors.

11.1.3 What is the relationship between the learning constructs measured by the observed assessment measures and those constructs represented by the detectors of student knowledge, behavior, and affect?

Many of the detectors of student knowledge, behavior, and affect were developed in consideration of one or more of the observed assessment measures. It is not clear, however, how the underlying constructs relate to each other across these detectors and assessment measures.

11.1.4 Which constructs represented by the detectors of student knowledge, behavior, and affect are reliable predictors of short- and long-term outcomes?

The reliability and predictive power of identified learning constructs measured by the detector models can be used to better understand their relationship with the observed learning outcome measures.

11.2 Detectors and Outcome Measures

11.2.1 Detectors of Student Knowledge

In the last two decades of research pertaining to learning analytics and educational data mining, the field has produced numerous models attempting to quantify student knowledge. By observing student correctness on problems within and across knowledge components, one can gain an understanding of how well students have seemingly mastered such content based on their predicted ability to answer future problems of the same skills. As such, a large subgroup of these fields of educational research has emerged surrounding the prediction of student 'next problem correctness'. While this endeavor holds little practical significance in terms of developing learning interventions (as recent attempts to improve such models have proven to yield only marginally small improvements), such models may be utilized for their original purpose of measuring student knowledge.

Among these models of student knowledge, however, few have been as arguably pivotal as the bayesian knowledge tracing model [CA95]. Among the four learned parameters of the model, two attempt to quantify each student's knowledge state as the prior knowledge and current level of mastery (both as binary learned and

unlearned state representations). Using bayesian models, researchers can look at student answers and estimate the student's knowledge state or if it is instead attributable to the student guessing at the answer (e.g. answering correctly despite not knowing the material) or slipping (e.g. knowing the material yet answering incorrectly) as a probability.

Another popular knowledge model, Performance Factors Analysis [PJCK09], takes a similar next-problem correctness approach though with a much simpler logistic-regression-based methodology. Unlike BKT, the PFA model observes just the number of correct and incorrect responses of students to construct a model of how likely a student is to correctly answer a problem of a given knowledge component. In the case of this model, this probability can be used to describe the knowledge level of the student.

More recently, deep learning methods, describing a family of techniques utilizing multi-layered neural networks, have exhibited an increase in usage in a wide-range of fields. This can be attributed to increased development support, advances in technology, and subsequently promising performance when compared to more traditional methods. A type of deep learning model, known as a recurrent neural network [WZ89], has been the basis of several recent works that suggest notable improvements to estimating short-term student performance. The development of the Deep Knowledge Tracing (DKT) model [PBH$^+$15] was among the first applications of this type of deep learning model within an educational context, reporting vast improvements over the widely applied models of BKT and PFA. While others found that these improvements were largely overestimated [KLM16][XZVIB16], the method still shows promise in its ability to model student knowledge over time.

While the DKT model is described here in an effort to comprehensively describe widely-cited models of student knowledge, this particular model is not observed

alongside the other detectors in this work. While the two models of BKT and PFA are intended to measure the same outcomes of knowledge through next problem correctness, DKT was omitted from the analyses of this work. This decision was also made considering that the DKT model represents student knowledge as a 200-value vector and would therefore detract from the interpretability of the analyses; the study of DKT's representation of student knowledge has been explored in prior work [YY18], and the inclusion of this model in similar analyses as described in this paper is planned for future work.

11.2.2 Detectors of Student Behavior

A large amount of previous research has focused on modeling student behavior in computer-based systems. One of the most informative forms of data that can be provided to teachers is not the end result or performance metric alone, but data that can describe the process that contributed to a result. As such, detectors of student behavior have emerged in the field of learner analytics and, among other systems, their application and further development have been studied using ASSISTments data.

Whether or not a student is attending to a learning task, described as on- and conversely off-task behavior, can help to distinguish levels idleness and critical thinking. From the point of view of the learning software, it is only known that a student is engaged when taking action within the system; it is the periods between such action, however, where learning occurs. Previous work has observed student on/off task behavior [BRX07][PBSP+14] in an effort to identify when a student actively engaging in the learning task. In addition to this detector, this work proposes to extend this detector by similarly interacting estimates of student on/off task behavior with a measure of time on task.

One of the more negative behaviors that has been studied is that of students "gaming of the system," or cheating the system, referred to hereafter simply as "gaming" [BCKW04][PBdCO15]. Student gaming is exhibited in a number of ways depending on the type of assignment and availability of computer-provided tutoring. This behavior can be described as a student progressing through an assignment by exploiting an aspect of the system rather than administering effort to learn the material. In such cases, the student may proceed quickly through the assignment, exhausting all computer-provided tutoring to reveal the correct answers (it is common to see these students finishing such assignments in just a few minutes' time, while the rest of the class takes significantly longer depending on the difficulty and number of questions). Developments toward detecting this behavior can help inform teachers that a student has not applied effort and likely does not know the assigned material despite having "completed" the assignment.

Another detector of student behavior observes a measure of student carelessness [BWH+08][PBBH13] during a learning task. This detector is based on the concept of student "slip" popularized by the bayesian knowledge tracing model. Carelessness is described as contextual slip, or the likelihood that, for a given problem, the student answered incorrectly despite having mastery of the material. Conversely, the detector of contextual guess is intended to measure the opposing case of a student answering correctly despite not knowing the material.

11.2.3 Detectors of Student Affect

Students' emotion and affective state have been proven as significant predictors of short- and long-term performance [CGSG04][PBSP+14]. Using student affect detectors researchers have reliably predicted affect from ASSISTments logs have used estimates affective state to better predict state test scores [PBSP+14], college

attendance [PBBH13], STEM-related college majors [SPOBH14], and how these detectors generalize across rural, urban, and suburban contexts [OBG+14]. With such works pointing to the importance of detecting and measuring student affect, the argument for their inclusion in this proposal is well-founded in this prior research.

A significant amount of research has been conducted on the detection of student affect state by aligning ASSISTments data to collected quantitative field observations using the Baker Rodrigo Ocumpaugh Monitoring Protocol (BROMP) [OBR15]. This protocol allows human coders to observe students in the classroom while working within the learning system and label them based on one of four commonly studied affective states: engaged concentration [Csi90], frustration [KRP01][PSC93], boredom [Csi90][Mis96], and confusion [CGSG04][KRP01].

Initial development of sensor-free affect detectors, utilizing only the recorded student log data aligned with human-labeled observations, explored a number of tree-based, rule-based, and Bayesian models, ultimately reporting moderate model accuracy above chance [OBG+14]. Later,[WHH15] improved upon these initial affect models by incorporating more information pertaining to skill, or knowledge component, as well as class-level features. Most recently, Heffernan and colleagues [BBH17] explored the application of deep learning models, exhibiting a significant increase to model performance. That work compared three variants of recurrent neural networks - traditional recurrent, LSTM, and Gated Recurrent Unit networks - as sequence-to-sequence models to estimate labeled student affect states.

A comprehensive list of included detectors, sorted by the identified folksonomy, is provided in Table 11.1.

Category	Detector
Detectors of Student Knowledge	BKT Knowledge Estimate
	PFA Knowledge Estimate
Detectors of Student Behavior	Off-Task
	Gaming the System
	Wheel Spinning
	Stopout
	Contextual Slip
	Contextual Guess
Detectors of Student Affect	Engaged Concentration
	Boredom
	Confusion
	Frustration

Table 11.1: The list of observed detectors of student knowledge, behavior, and affect.

11.2.4 Measures of Unproductive Perseverance

Several previous works have attempted to model student wheel spinning behavior in several platforms including Cognitive tutor [MCS16] and ASSISTments [BG13], while other work has explored policies to help prevent wheel spinning [GB15]. As previously described, wheel spinning is the behavior in which a student exhibits high persistence in a learning task, but unable to obtain sufficient understanding of the learning materials. The term "wheel spinning" is analogous to a car that is stuck in snow or mud; despite devoting effort into moving, the wheels will spin without getting anywhere.

In this work, we will be using the definition of wheel spinning given in [BG13] as failing to reach mastery after seeing ten learning opportunities. It is for this reason that prior work observing wheel spinning has pertained to student interactions with mastery-based assignments. Mastery-based assignments, as opposed to traditional assignments that require students to answer all assigned problems, instead require students to demonstrate a sufficient level of understanding, or mastery, of the assigned material in order to complete the assignment. In the case of AS-

SISTments, this threshold of understanding, by default, requires students to simply answer three consecutive problems correctly on the first attempt without the use of computer-provided aid.

Conversely from that of wheel spinning, student attrition, characterized by a student failing or refusing to complete a given assignment, describes cases of low persistence. In the context of K-12 classrooms, it is common to observe student attrition at the assignment-level, where students begin an assignment but fail or choose not to complete the assigned work. This behavior, which has been referred to as "stopout," [BVIH19] is distinctly different from the course-level dropout that is observed in MOOCs as students likely return to work on subsequent assignments; the student remains in the course, but did not finish the assigned work. As is the definition used in [BVP$^+$ss], stopout is defined as mutually exclusive to that of wheel spinning as it is considered desirable to stop attempting problems once wheel spinning behavior has been exhibited; as such, stopout is defined as the lack of assignment completion before the 10th problem, or learning opportunity.

Similarly, it was identified in [BVIH19] and further explored in [BVP$^+$ss] that stopout exhibited early in an assignment is seemingly different from that of later stopout and includes a behavior identified as "refusal," particularly when such attrition occurs on the first problem. As such, stopout will further be broken into two categories corresponding to early stopout (lack of completion on or before the third learning opportunity) and later stopout (lack of completion between the fourth and 9th opportunities inclusively).

While the particular focus of this work is on outcomes pertaining to productive and unproductive perseverance, additional measures are observed to gain understanding of relationship between the learning constructs represented by the observed detectors and other outcomes that describe student learning. Among these

Outcome Measure	Description
Wheel Spinning	Whether the student fails to complete a skill builder assignment on or before the 10th learning opportunity
Early Stopout	The student fails or refuses to complete the assignment on or before the 3rd learning opportunity
Later Stopout	The student fails or refuses to complete the assignment between the 4th and 9th learning opportunities, inclusively
Next Problem Correctness	The binary correctness of the students first response on the next problem (with a help request treated as incorrect)
Next First Action	Whether the first action on the next problem will be an attempt to answer (=1) or a help request (=0)
Assignment Completion	Whether the student completes the current assignment
Number of Problems	A simple count of the number of problems attempted by the student on the assignment
Inverse Mastery Speed	The inverse of the number of problems needed to correctly answer 3 problems correctly in a row without the use of computer-provided aid (or 0 when this threshold is not met)
TerraNova Score	The end-of-year standardized test score available for a subset of students

Table 11.2: The description of outcome measures

additional outcomes that include short-term performance measures of next problem correctness and next first action, perhaps most notable is the outcome of TerraNova Score. This distal measure represents student performance on an end-of-year standardized assessment; the inclusion of such a measure, particularly as it was delivered externally to the observed learning platform of ASSISTments, can act as a measure to externally validate the importance of constructs underlying the observed detectors. A comprehensive list and short description of each included outcome measure is provided in Table 11.2.

11.3 Methodology

11.3.1 Data

The data[1] used in the methods described in this work consists of three datasets collected through the ASSISTments online learning platform; the scale of this data is described in Table 11.3. ASSISTments is a web-based learning platform that provides the tools for teachers to assign classwork or homework content for which students receive immediate correctness feedback [HH14]. While working through each assignment, many problems supply students with optional aid in the form of either hints or scaffolding questions. Hints, of which there may be from 0 up to several available, supply students with an instructional message, while scaffolding, when available, breaks the problem into smaller steps to solve. In addition to these, the system provides a "bottom-out" hint for every problem that supplies the students with the correct answer if the student is unable to solve the problem as, by default, students are not allowed to progress to subsequent problems until the correct response is entered inside ASSISTments. The analyses described in this work includes only student interaction data with mastery-based assignments, known as "skill builders" in the system, where the completion threshold is designated to simply require students to answer three consecutive problems correctly without the use of computer-provided aid (i.e., without hints, scaffolding, or bottom-out hints); students are continually presented with problems consisting of a single or a small number of related knowledge components until this completion threshold is met. In recognition of wheel spinning as an undesirable learning behavior, the system implements a "daily limit," stopping students on the skill builder assignment for the day if the completion threshold is not reached by the tenth problem (except in the case

[1] All data and code used in the analyses described here are made openly available at the following link: `http://tiny.cc/dissertation_data`

	Dataset 1 (Remnant)	Dataset 2	Dataset 3
Number of Students	30,266	16,504	12,105*
Number of Assignments	9,284	3,420	3,349
Number of Classes	1,212	641	395

*663 Students across 36 classes have an associated TerraNova Sore

Table 11.3: Counts of students, assignments, and classes across the three datasets.

where the student is about to reach the threshold on or directly following the tenth problem); the system provides the student with an instruction to seek additional help and return to the assignment on the subsequent day.

As teachers using the system assign a range of content, both made available through the system as well as self-built material, all datasets used in this work include skill builder assignments where at least 10 students started the assignment and the overall completion rate is at least 70% on the assignment within the class. These limitations help to remove outliers such as sample classes and optional supplementary assignments where the teacher does not expect and require every student to complete. These outlier cases are excluded as we would argue that attrition due to such factors is not stopout as we have defined it within this task (e.g. low unproductive persistence). This filtering process is the same as was used in prior work [BVIH19] characterizing early stopout behavior.

The usage of three datasets serves the purpose of providing sufficient held-out data for the purpose of validating the methods described in the next section. The first dataset, for example, consists of student data from the 2012-2013, 2013-2014, and 2018-2019 academic years. This dataset, henceforth called the "remnant (named after the remnant defined in [SBPH18]used to describe students from outside each experiment), contains the student data necessary to train each of the detector models listed in Table 11.1. For example, the data used to train the models of affect, off-task behavior, and gaming observed in [BBHon] is a subset of the remnant as these models

were again applied in the current work. The sole purpose of this remnant dataset is for model training, development, and evaluation (as in the case of a few models for which new modeling methods were applied as is described in the next section), after which such models are subsequently applied to the other two datasets used in this work. While the remnant does contain some overlap with our third dataset in regard to which academic years of students are represented (described later in this section), it is important to emphasize that there is no overlap in regard to students across the three datasets used in this work.

Once the detector models have been trained and evaluated using the remnant, they are then applied to the second dataset and used for the next set of analyses in which a factor analysis is applied across the generated detector estimates. This second dataset consists of student skill builder data from the 2016-2017 and 2017-2018 academic years. It is with this dataset that a factor analysis is conducted across the detector estimates to identify the underlying represented constructs. The dataset is scaled to include the two aforementioned academic years in an attempt to maximize the representative populations of students to benefit the extrapolation of analyses and models to new students from differing academic years.

The third and final dataset consists of student skill builder data from the 2013-2014 and 2014-2015 academic years and includes only those students involved in the ASSISTments efficacy trial conducted within the state of Maine [RFMM16]. It is with this final dataset that the final set of analyses are conducted to validate the factor analysis applied to the second dataset using a confirmatory factor analysis, explore the relationship between underlying constructs using an exploratory structural equation model, and finally examine the relationship of the measured constructs on short and long-term learning outcomes. Each student in this dataset, in addition to the interaction data collected within ASSISTments, also has an associ-

ated TerraNova state test score given at the end of the academic year. Teachers and their students participated in the study for a single year within one of two cohorts, each following seventh grade mathematics curricula across a variety of textbooks that were incorporated into ASSISTments (i.e. allowing teachers to assign the same content through ASSISTments that they had in previous years using traditional methods). This aspect is important to specify as teachers were certainly not required to assign skill builders during this study, however a majority did use such assignments regularly, providing a sufficiently-scaled dataset to conduct the analyses described in this work.

11.3.2 Detector Models

Many of the detectors utilized in this work were previously developed in previous works using ASSISTments data (some of these models, such as gaming, were first developed in another system and then appropriated and re-fit using ASSISTments data [PBdCO15]). Of the detectors listed in Table 11.1, the detectors of student affect, off-task behavior, and gaming were directly applied from prior work comparing the use of machine-learned and expert-generated features [BBHon]; the best performing detector model of each was trained and applied as was described in that prior work, utilizing recurrent long short term memory (LSTM) networks, co-trained using labeled and unlabled data. This, again, was made possible as the training data utilized in that work was incorporated into the remnant used here. All models used in this work with the exception of the BKT and PFA models were applied to data separated into approximately 20 second clips of student activity. This follows the same methodology as was applied in prior works [OBG$^+$14][BBH17][BBOH18]. The BKT and PFA models were applied at the problem-level, and the estimates of each other detector are later aggregated to this problem level for subsequent analyses.

The remaining detector models, however, needed to be either trained or re-fit using the remnant dataset. For the BKT model and PFA models, for example, this required simply training each of these on the specific skills contained within the remnant. The structure of each of these models is well-defined by prior works; one model per skill was created for each BKT and PFA using the basic formulation of each model (e.g., while some extensions to BKT have been proposed in [PH10a][PH11][WH13], only the traditional 4-parameter BKT model was applied in this work). It was found that not all problems in the dataset contained a skill-tagging. For these cases, a skill tag was either generated using the common core state standard [A$^+$10] identifier often contained within the name of each skill builder, or labeled as "no skill" for any remaining cases. This step also helps to extrapolate these models to the other datasets which may contain data from skills that did not exist in the remnant; such cases could simply use the "no skill" model as a noisy estimate of student knowledge as it is averaged across multiple unlabeled skills. Once trained, the knowledge parameter of the BKT model is used when applying the model to the subsequent datasets. Similarly, the PFA model is trained on the same set of skills and the models estimate of next problem correctness is used as a measure of student knowledge for subsequent analyses.

The detectors of contextual guess and slip had previously been fit to ASSISTments data [PBSP$^+$14][PBBH13], but were readdressed in this work following the successful application of deep learning methods for similar tasks in previous works [BBH17][BBOH18][BBHon]. For this task, the same methodology as is described in [dBCA08] is utilized in the current work to generate labels of contextual guess and slip using triplets of estimates from the previously-described BKT model; the student performance on a given problem and the two subsequent problems is used in conjunction with the parameter estimates from BKT to label each problem with

a probability label corresponding with either guessing (when the student answered the problem correctly) or slipping (when the student answered the problem incorrectly). With these probability labels, a deep learning model is applied following the same structure as the detector models described in [BBHon] and using the same set of 92 features as utilized in that work. The models used the same structure with only two small differences: 1) 50 hidden nodes were used in place of 200 (chosen by comparing each of these models using a small hold-out set), and 2) as the previous work observed binary labels and the contextual guess and slip are continuous values, a linear output function was used in place of the softmax output utilized in the prior work. A separate model is trained to predict each contextual guess and slip, producing a model estimate for each clip of student activity in the remnant. The models were evaluated within the remnant using a 10-fold cross validation resulting in R2 values of .391 and .301 for the models of contextual guess and slip respectively.

Previous work explored the development of early detectors of student stopout and wheel spinning [BVP$^+$ss]. The purpose of that work, however, was to explore aspects of model transfer to identify the commonality of machine-learned features for predicting each of stopout and wheel spinning. As such, these models are refit in this current work with the intention of using them as predictive models of these behaviors. Similar to the models of contextual guess and slip, the same model structure as was used for the detectors described in [BBHon] were used for consistency. While two small changes were described for the contextual guess and slip models, it was found that the exact same model structure as the off-task detector from [BBHon] led to the better performing models when applied to a small holdout set (after similarly testing a smaller hidden layer). A model was trained for each stopout and wheel spinning using the same definition of these behaviors as was used in [BVP$^+$ss], where wheel spinning describes students who have not mastered a skill

builder on or after the tenth problem and stopout is mutually exclusive describing students who do not complete the assignment before the ninth problem (it is important to emphasize that stopout is further divided in later analyses conducted here as expressed in Table 11.2. These models were similarly evaluated using a 10-fold cross validation within the remnant and exhibited ROC AUC values of 0.878 and 0.731 for wheel spinning and stopout respectively. Once evaluated, the models are trained on the full remnant dataset and then applied to the second dataset as was done for each other detector.

11.4 Factor Analyses

An exploratory factor analysis (EFA) was applied to address the first research question focused on identifying the underlying constructs measured by the observed outcomes listed in Table 11.2. Using the second dataset, all outcomes were observed at the problem level. As the granularity of outcomes varied (e.g. next problem correctness compared to assignment completion), all outcomes were observed at the problem level, with higher-level outcomes being represented as duplicated values for each problem. TerraNova score was not included in this analysis as 1) no students in the second dataset had associated TerraNova scores, but also 2) there would likely be a large amount of noise in comparing problem-level outcomes to an end-of-year assessment. The two remaining non-binary outcomes of Number of Problems and Inverse Mastery Speed were transformed into approximated normal distributions for the purpose of the EFA and normalized using z-scoring.

In applying the EFA, a maximum likelihood extraction method was used with an oblimin rotation; these allow for correlated factors to be extracted and are common choices when performing a EFA such as this. Allowing for correlated factors in this

	Factor 1 (Wheel Spin)	Factor 2 (Completion)	Factor 3 (Later Stopout)	Factor 4 (Early Stopout)
Number of Problems	1.0			
Assignment Complete		0.93		
Early Stopout				-0.63
Later Stopout			-0.72	
Next First Action				
Next Problem Correct	-0.44			
Wheel Spin	0.71			
Inverse Mastery Speed	-0.83			
Variance Explained	30.7%	12.7%	8.6%	6.3%

Table 11.4: The EFA factor loadings observing the student learning assessment measures.

particular analysis is important as it is likely that many learning constructs exhibit relationships. The number of factors is determined using a parallel analysis [Hor65].

The results of the EFA are reported in Table 11.4. From this table, it can be seen that 4 factors emerge from the outcome labels. In this regard, the second, third, and fourth outcomes each exhibit a single aligned label, and are therefore referred to by the aligned labels of Assignment Completion, Later Stopout, and Early Stopout respectively (with the later two found to inversely represent the factors). The final factor, represented with a high number of problems, wheel spinning, and inversely next problem correctness and inverse mastery speed, appears to highly relate to the definition of wheel spinning; students are attempting a large number of problems with a low percent correct (demonstrating low knowledge). It is also important to recognize that the outcome of Next First Action did not align highly to any of the four factors.

An additional EFA was applied to address second research question focused on identifying the underlying constructs measured by the observed detectors listed in Table 11.1. Using the second dataset to which each of the detectors were applied, each detector was aggregated to the problem-level using a simple average; this en-

	Factor 1 (Negative Affect)	Factor 2 (Carelessness)	Factor 3 (Knowledge)	Factor 4 (Disengagement)
Confusion	0.96			
Concentration	-0.98			
Boredom	0.81			0.45
Frustration	0.99			
Off Task				0.92
Contextual Guess		-0.94		
Contextual Slip		0.88		
Stopout		0.43		
Wheel Spin		0.50	-0.43	
Gaming				0.31
BKT P(Know)			0.83	
PFA			0.75	
Variance Explained	30.3%	17.9%	13.0%	10.8%

Table 11.5: The EFA factor loadings observing the detector models.

sures that there is a single estimate per detector for each problem started by each student. Aggregation to this level ensures consistency in regard to the granularity of each of the detectors as well as better alignment to the most fine-grained outcomes observed (e.g. next problem correctness and next first action). After aggregation, several simple transforms were applied to the detectors so that they follow a more approximate normal distribution as there are several advantages to then when applying EFA. All estimates were then z-scored within their respective detector to standardize their values.

Similar to the previous EFA, maximum likelihood with oblimin rotation was used to extract factors. These factors are reported in Table 11.5. The number of factors was determined using the same method as the previous EFA conducted over the observed outcomes, and similarly found 4 factors represented by the detectors. In this case, none of the factors were represented by a single detector, suggesting that groups of detectors are measuring a common set of underlying learning constructs in potentially different ways. While the current work does not attempt to measure the degree to which these detectors overlap, such an analysis would be worth attention

in future work.

The first factor identified consists of the four estimates produced by the affect detector model, with confusion, boredom, and frustration aligning positively, and concentration aligning negitively. Such an alignment loosely corresponds with aspects of disequilibrium hypothesized by DMello and Graesser (2012) [DG12], but as the alignment also suggests a level of disengagement, this first factor will simply be referred to as Negative Affect.

The second factor aligns highly with the contextual guess and slip (negatively and positively, respectively), as well as both stopout and wheel spinning. It is important to emphasize here that the detectors, while correlated with the respective labels on which theyve been trained, are still an estimate of future performance (i.e. the wheel spinning detector should not be interpreted as strongly as the true label of wheel spinning). With this in mind, especially as stopout and wheel spinning both positively align with this factor and represent conflicting measures of persistence, it is believed that this factor is instead more representative of poor student performance (as a contextual slip suggests that the problems are answered incorrectly). As such, this factor will be referred to as Carelessness, following the alternative name given to the contextual slip detector in previous works [SPdBR11].

The third and fourth factors are arguably the most identifiable factors that emerged. In the case of factor three, for example, where the two detectors of student knowledge aligned positively and wheel spinning aligned negatively, we conclude with confidence that this factor is a representation of student knowledge and will therefore be referred to as simply Knowledge. Finally, the last factor exhibited positive alignment from the off-task, boredom, and gaming detectors. In this regard, each of these represent a level of low student effort and disengagement from the learning task. As such, this factor will be referred to as Disengagement.

11.5 The Relationship Between Factors

With the EFA applied to the second dataset as described in the previous section, the third and fourth research questions are addressed using the third and final dataset. First, it is important to ensure that the alignment of factors is consistent between the second and third datasets. For this reason, a confirmatory factor analysis (CFA) is applied using the factor loadings from the previous EFA conducted across the detector models. The model learned in the first EFA (in regard to the magnitude and direction of factor loadings) is compared for goodness-of-fit within the third dataset. Four commonly-reported CFA metrics are reported in Table 11.6. While much of the prior work surrounding these metrics have relied on rules of thumb to determine what constitutes a "good" fit, low values of RMSEA and SRMR are considered better while high values (close to 1.0) of CFI are prefered. Another commonly reported metric of a p-value calculated from a chi-squared analysis was omitted from this analysis as it is sensitive to large sample sizes (the metric will often be significant with large sample sizes, which was the case when applied here, but this offers little further insight beyond the other metrics). These values do fall within the commonly-accepted range of fit and therefore support that the model does generalize across the two datasets.

RMSEA	CFI	SRMR
0.107	0.944	0.050

Table 11.6: The CFA measures of fit for the EFA observing the detector models.

A similar analysis was conducted on the EFA considering the outcome labels as well with considerably lower goodness-of-fit; these metrics are reported in Table 11.7. The poorer metrics suggest that the outcome labels are measuring either different factors, or, as is hypothesized here, are measuring the factors in slightly different

ways. This may be a result of differences in how teachers participating in the ASSISTments efficacy trial assigned skill builder assignments (as they were often in accompaniment with other textbook work), but it is important to emphasize that this is merely speculation.

RMSEA	CFI	SRMR
0.351	0.506	0.174

Table 11.7: The CFA measures of fit for the EFA observing the student assessment measures.

In order to address the third research question, we apply a ESEM to observe the correlational and suggested causal relationships that emerge between the factors identified from the EFAs conducted over the detectors and outcomes. The result of this ESEM is illustrated in Figure 11.1.

From this, it is visible that the same factors emerge for the detectors, but there are some differences in the emerging factors underlying the outcomes (as was suggested by the CFA). Here, it can be seen that the same factors identified as Wheel Spinning and Early Stopout do emerge, but two other factors emerged that were not present in the previous EFA (corresponding to Next Problem Correctness and additional loadings of outcomes alongside Assignment Completion). As these did not generalize across the datasets, no strong claims can be made regarding these additional factors. However, as the Wheel Spinning factor did emerge in both sets, the ESEM does suggest that there is a relationship between the factors of Knowledge (negatively) and Carelessness (positively) identified from the detector EFA. A negative relationship is also found between the Knowledge and Carelessness factors. Intuitively, these relationships do make sense, particularly in the context of the definition of wheel spinning. Generally this behavior is characterized as a student exhibiting low knowledge and struggle (as seemingly captured by the factor

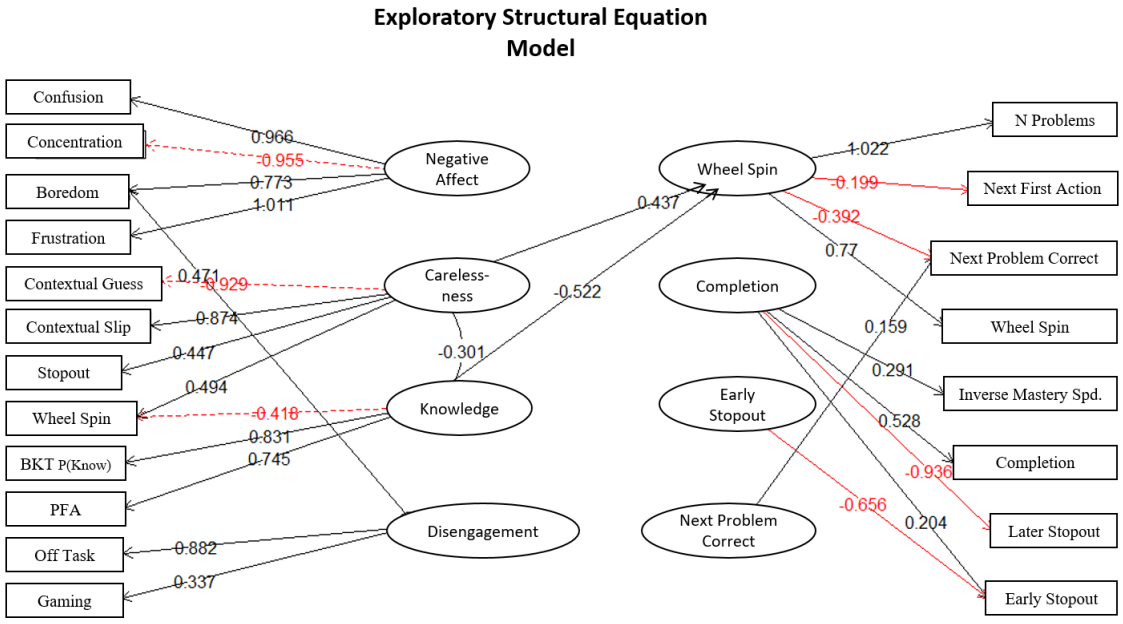

Figure 11.1: The resulting structural equation model applied to the third dataset.

being referred to as Carelessness) which causes a high number of problems to be attempted, low correctness, a need to request help (i.e. in the case of Next First Action exhibiting a negative relationship) and, of course, wheel spinning behavior.

11.6 Predictive Models of Unproductive Perseverance and Performance

To address the final research question, the learning constructs represented by the detectors of student knowledge, behavior, and affect are observed in relation to each observed learning outcome. Particular focus, however, is given to the two outcomes believed to most closely measure unproductive perseverance: assignment wheel spinning and early stopout.

To achieve this, the factors developed from the detectors in the second dataset are extracted as additional features in the third dataset using a linear combination

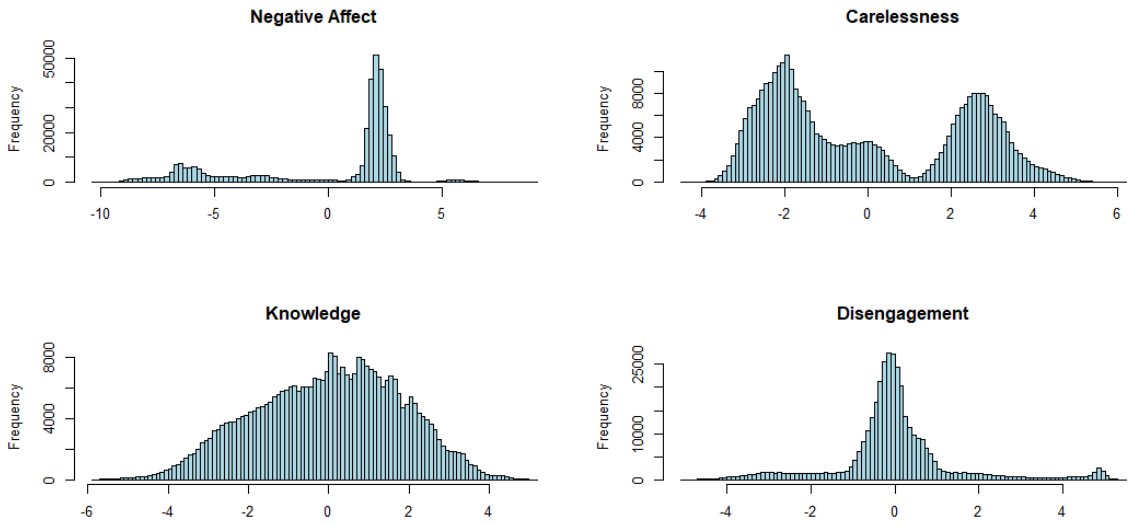

Figure 11.2: The distributions of each factor extracted from the detector models.

of detector estimates (based on factor loadings). The distributions of these four values is illustrated in Figure 11.2. From this, it can be seen that Carelessness and, arguably, Negative Affect exhibit bimodal distributions. As such, two new binary features are generated to indicate high and low modalities for each factor estimate using 0 and 1 as the cut points for Negative Affect and Carelessness respectively. With these points, each of these factors are z-scored within sub-distribution, collapsing the values into an approximately-normal distribution. These transformed features were then used in conjunction with the binary feature indicating high and low sub-distributions as well as their interaction to capture these groupings. Knowledge and Disengagement are then z-scored as well such that each of the factors are standardized for use within the set of predictive models. The transformed distributions are illustrated in Figure 11.3.

A 2-level hierarchical linear model is used to model each of the observed outcomes. As each of the factor estimates exist at the problem-level, each of the models observes sets of problems nested within student. The model structure was deter-

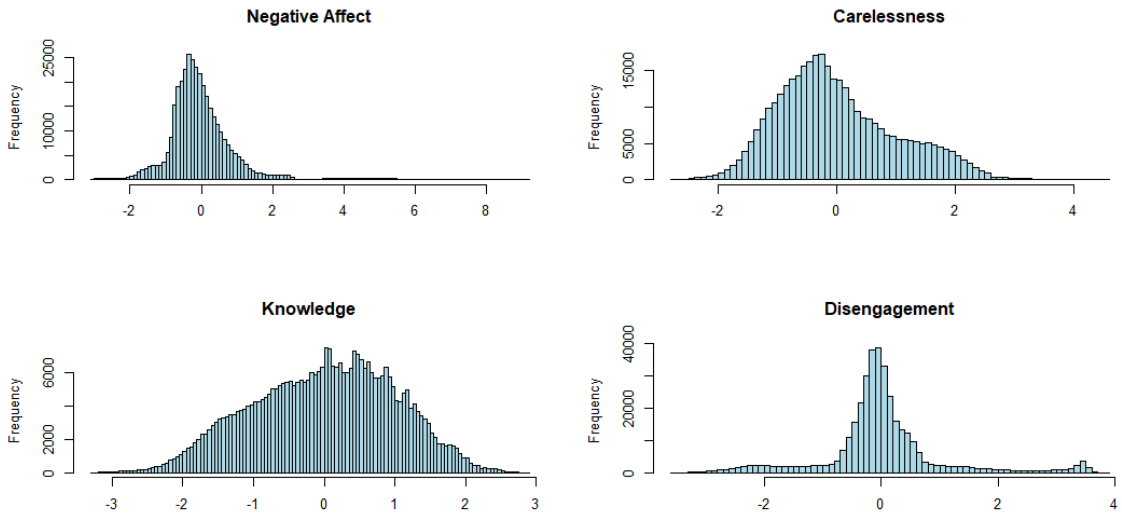

Figure 11.3: The transformed distributions of each factor extracted from the detector models.

mined by observing the intraclass correlation of the null model (i.e. the multi-level model fit only with intercepts) to compare a 2- and 3- level model with students additionally nested within classes; it was found that the ICC did not suggest that a 3 model was necessary and there were a relatively small number of distinct classes (n=36) on which to fit a multi-level model. The precise formula used for each model, with the exception of the model predicting TerraNova score, is expressed in the following equation:

$$
\begin{aligned}
Outcome_i = {} & \beta_1 * NegativeAffect + \beta_2 * Carelessness + \beta_3 * Knowledge \\
& + \beta_4 * Disengagement + \beta_5 * HighNegativeAffect \\
& + \beta_6 * HighCarelessness \\
& + \beta_7 * NegativeAffect \times HighNegativeAffect \\
& + \beta_8 * Carelessness \times HighCarelessness \\
& + (1|Student)
\end{aligned}
$$

Where the final term denotes a random intercept fit for each student. As all but the Number of Problems and Inverse Mastery Speed labels are binary, such labels are represented using logit-based functions while the two non-binary labels are modeled as linear; the transformed, normalized versions of Number of Problems and Inverse Mastery Speed are used such that they follow an approximated-normal distribution.

The beta coefficients for each of the predictors as well as the R^2 values for each model are reported in Table 11.8. The R^2 appears as two values for the purpose of the multi-level model, calculated using the "theoretical" method used in Nakagawa et al., 2017 [NJS17]; the first value represents the variance explained by the fixed effects (i.e. the factors extracted from the EFA applied to the detectors), while the second value is a cumulative variance explained by the entire model (e.g. fixed effects plus random effects).

	Wheel Spin	Early Stopout	Later Stopout	Next First Action	Next Problem Correctness	Assignment Completion	Number of Problems	Inverse Mastery Speed
Intercept	-4.86***	-6.05***	-6.82***	3.51***	1.34***	5.67***	-0.54***	0.49***
Negative Affect	0.04**	-0.10*	0.06*	-0.15***	-0.02**	-0.04*	0.01***	-0.03***
Carelessness	0.84***	0.17**	0.31***	-0.24***	-0.07***	-0.37***	0.28***	-0.28***
Knowledge	-0.60***	-0.59***	-0.64***	0.35***	0.48***	0.79***	-0.22***	0.24***
Disengagement	-0.40***	0.34***	-0.01	0.06***	-0.00	0.02*	-0.14***	0.12***
High Negative Affect	0.94***	-0.31***	0.12***	0.01	0.05***	-0.09***	0.35***	-0.33***
High Carelessness	1.52***	0.74***	1.03***	-1.52***	-0.51***	-1.19***	0.52***	-0.59***
High X Negative Affect	0.22***	-0.18**	-0.15***	0.07***	0.01	0.01	0.10***	-0.04***
High X Carelessness	-0.88***	0.13	-0.14**	0.13***	-0.04**	0.24***	-0.30***	0.26***
Level 2 ICC	0.79	0.70	0.84	0.30	0.09	0.82	0.32	0.37
R^2	0.15 / 0.80	0.10 / 0.72	0.05 / 0.85	0.18 / 0.39	0.11 / 0.12	0.08 / 0.84	0.34 / 0.50	0.33 / 0.54

Table 11.8: The beta coefficients, statistical reliability, and variance explained for each multi-level model.

In first observing the resulting models for the outcome measures of Wheel Spinning and Early Stopout, it can be seen that the predictors account for 15% and 10% of the variance respectively. However, in either case, the random effects (e.g. those explained by the student level of the model) explain a much larger proportion of variance. In the case of Wheel Spinning, the largest positive predictor is that of the binary indicator of high carelessness, representing the higher-valued subdistribution. Similarly, the continuous-valued carelessness factor along with the indicator of high negative affect all positively correlate with wheel spinning behavior. What is also notable, aside from the unsurprising negative correlation of knowledge, is the similarly negative correlation of disengagement. As disengagement is positively correlated with early stopout, students are less likely to persist when exhibiting this identified construct. Observing further differences between the models observing Wheel Spinning and Stopout, the role of Negative Affect appears to be inverted. Higher values of negative affect appear to correlate negatively with early stopout; in other words, higher estimates of negative affect correlate with persistence beyond the third problem. In this sense, this distinction appears to distinguish negative affect from disengagement. By further observing the label of Later Stopout, it would appear that the identified construct of disengagement is attributable to early stopout but then shifts where carelessness is then the more correlated factor with later stopout and wheel spinning.

The subsequent models do also illustrate some relationships that are worth identifying. Unsurprisingly, knowledge has a strong positive relationship with Next First Action, Next Problem Correctness, and Assignment Completion. The binary indicator of high carelessness also appears to have a strong negative relationship across all the models. Also rather surpising is that the model of Next Problem Correctness exhibited the lowest amount of variance explained; furthermore, the low ICC

	Estimate	Std. Error	p-value
Intercept	0.057	0.151	0.707
High Negative Affect	-0.058	0.038	0.130
Low Negative Affect	-0.115	0.037	0.002**
High Carelessness	0.061	0.037	0.104
Low Carelessness	-0.191	0.055	<0.001***
Knowledge	-0.129	0.067	0.060
Disengagement	-0.060	0.041	0.150
Level 2 ICC	0.625		
R^2	0.025 / 0.634		

Table 11.9: The model results observing the distal outcome of TerraNova score.

suggests that there is little variance being explained at the student level for this label.

11.6.1 Observing Distal Student Performance

The final model utilizing the extracted factors underlying the detectors of student knowledge, behavior, and affect observes the distal outcome of TerraNova score. Again, this score is an end-of-year assessment that was completed by students outside the learning platform of ASSISTments. As such, it acts as a truly external measure on which to explore the identified constructs. While the previous set of models observed shorter-term outcomes, it is likely unreasonable to expect a similar model to predict end-of-year test scores at an individual problem level without capturing noise. As such, the extracted estimates are aggregated to the student level. In the case of the two bimodal factors, each are separated and aggregated separately as a "high" and "low" measure when averaged across each student's sequence of problems solved over the year. A 2-level linear model is fit to the data observing class as the second level (as the first level now represents a single student). Each of the aggregated factors as well as TerraNova score are z-scored to produce standardized coefficient estimates.

The results of this model are reported in Table 11.9. From this, it is found that only two of the aggregated factors are found to be statistically reliable. The values of low negative affect and low carelessness each exhibit negative correlations with the student TerraNova score. The estimate of knowledge is also suggestively positively related, with a p-value that is just above the threshold of 0.05. Overall, the fixed effects (i.e. the factors) account for only 2% of the variance as illustrated by the R^2 value, while an additional 60% is explained by the class level of the model.

11.7 Conclusions

Across the several analyses described in this work, the constructs of Negative Affect, Carelessness, and, to a lesser degree, Knowledge were consistently found to be predictive of student learning outcomes. Disengagement, while found to be reliable predictors of most observed outcomes, appeared to have the strongest relationship with early stopout and a negative relationship with measures of higher persistence such as wheel spinning. From these relationships, the results support the idea that these constructs are distinguishing productive and unproductive aspects of persistence. Future work can take these analyses a step further to look at interactions across these factors and identify potential groupings of students.

The results reported in Tables 11.8 and 11.9 do identify reliable relationships between the identified factors and commonly-observed outcomes, but in many cases there is a considerable proportion of variance left unexplained by the models. Furthermore, the outcomes of unproductive perseverance (i.e. wheel spinning and early stopout) exhibited a large proportion of variance explained at the student-level. The constructs explored in this work emerged from observing granular student interactions with the system. Future work could also focus on exploring learning constructs

that emerge at various levels of granularity in regard to the outcomes that are measured. In addition to this, as it was found that the factors underlying the observed outcomes did not generalize well to the final dataset, future work could focus on identifying teacher-level factors that may influence how such outcomes measure underlying factors.

This work represents an initial step toward furthering our understanding of the constructs of learning that emerge from the application of data-driven methods. Identifying and measuring the relationship between these constructs can help to guide future research toward developing interventions to address particular unproductive learning practices. Similarly, analyses such as those applied in this work help to identify how to best measure these learning constructs in order to look for differences that may occur from the application and deployment of directed learning behaviors.

Bibliography

[A+10] National Governors Association et al. Common core state standards. *Washington, DC*, 2010.

[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, ..., and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[AC06] Saleema Amershi and Cristina Conati. Automatic recognition of learner groups in exploratory learning environments. In *International Conference on Intelligent Tutoring Systems*, pages 463–472. Springer, 2006.

[ACB+09] Ivon Arroyo, David G Cooper, Winslow Burleson, Beverly Park Woolf, Kasia Muldner, and Robert Christopherson. Emotion sensors go to school. In *AIED*, volume 200, pages 17–24, 2009.

[AG02] Massih-Reza Amini and Patrick Gallinari. Semi-supervised logistic regression. In *ECAI*, pages 390–394, 2002.

[Ale01] Vincent Aleven. Helping students to become better help seekers: Towards supporting metacognition in a cognitive tutor. *German-USA Early Career Research Exchange Program: Research on Learning Technologies and Technology-Supported Education, Tubingen, Germany*, 2001.

[AMRK06] Vincent Aleven, Bruce Mclaren, Ido Roll, and Kenneth Koedinger. Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education*, 16(2):101–128, 2006.

[BBH17] Anthony F Botelho, Ryan S Baker, and Neil T Heffernan. Improving sensor-free affect detection using deep learning. In *International Conference on Artificial Intelligence in Education*, pages 40–51. Springer, 2017.

[BBHon] Anthony F Botelho, Ryan S Baker, and Neil T Heffernan. Machine-learned or expert-engineered features? exploring feature engineering methods in detectors of student behavior and affect. In Submission.

[BBOH18] Anthony F Botelho, Ryan S Baker, Jaclyn Ocumpaugh, and Neil T Heffernan. Studying affect dynamics and chronometry using sensor-free detectors. In *Proceedings of the 11th International Conference on Educational Data Mining*, pages 157–166, 2018.

[BCKW04] Ryan Shaun Baker, Albert T Corbett, Kenneth R Koedinger, and Angela 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*, pages 383–390. ACM, 2004.

[BCW06] Ryan SJd Baker, Albert T Corbett, and Angela Z Wagner. Human classification of low-fidelity replays of student actions. In *Proceedings of the Educational Data Mining Workshop at the 8th International Conference on Intelligent Tutoring Systems*, volume 2002, pages 29–36, 2006.

[BD13] Nigel Bosch and Sidney D'Mello. Sequential patterns of affective states of novice programmers. In *The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013)*, pages 1–10, 2013.

[BD17] Nigel Bosch and Sidney D'Mello. The affective experience of novice computer programmers. *International Journal of Artificial Intelligence in Education*, 27(1):181–206, 2017.

[BDRG10] Ryan SJd Baker, Sidney K D'Mello, Ma Mercedes T Rodrigo, and Arthur 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):223–241, 2010.

[BG13] Joseph E Beck and Yue Gong. Wheel-spinning: Students who fail to master a skill. In *International Conference on Artificial Intelligence in Education*, pages 431–440. Springer, 2013.

[BH95] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995.

[BM98] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[BmC07] Joseph E. Beck and Kai min Chang. Identifiability: A fundamental problem of student modeling. In *User Modeling 2007*, pages 137–146. Springer Berlin Heidelberg, 2007.

[BmCMC08] Joseph E. Beck, Kai min Chang, Jack Mostow, and Albert Corbett. Does help help? introducing the bayesian evaluation and assessment methodology. In *Intelligent Tutoring Systems*, pages 383–394. Springer Berlin Heidelberg, 2008.

[BOAss] R.S. Baker, J.L. Ocumpaugh, and J.M.A.L. Andres. Bromp quantitative field observations: A review. *R. Feldman (Ed.) Learning Science: Theory, Research, and Practice*, In Press.

[BP18] Nigel Bosch and Luc Paquette. Metrics for discrete student models: Chance levels, comparisons, and use cases. *Journal of Learning Analytics*, 5(2):86–104, 2018.

[BR14] Joseph Beck and Ma. Mercedes T. Rodrigo. Understanding wheel spinning in the context of affective factors. In *Intelligent Tutoring Systems*, pages 162–167. Springer International Publishing, 2014.

[BRX07] Ryan SJd Baker, Ma Mercedes T Rodrigo, and Ulises E Xolocotzin. The dynamics of affective transitions in simulation problem-solving environments. In *International Conference on Affective Computing and Intelligent Interaction*, pages 666–677. Springer, 2007.

[BVIH19] Anthony F Botelho, Ashvini Varatharaj, Eric G Van Inwegen, and Neil T Heffernan. Refusing to try: Characterizing early stopout on student assignments. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 391–400. ACM, 2019.

[BVP$^+$ss] Anthony F Botelho, Ashvini Varatharaj, Thanaporn Patikorn, Diana Doherty, Seth Adjei, and Joseph E Beck. Developing early detectors of student attritionand wheel spinning using deep learning. *IEEE Transactions on Learning Technologies Special Issue on Early Prediction and Supporting of Learning Performance*, In Press.

[BWH$^+$08] Ryan Baker, Jason Walonoski, Neil Heffernan, Ido Roll, Albert Corbett, and Kenneth Koedinger. Why students engage in" gaming the system" behavior in interactive learning environments. *Journal of Interactive Learning Research*, 19(2):185, 2008.

[BWH15] Anthony Botelho, Hao Wan, and Neil Heffernan. The prediction of student first response using prerequisite skills. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 39–45. ACM, 2015.

[CA95] A.T. Corbett and J.R. Anderson. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.

[Car97] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[CDWG08] Scotty D Craig, Sidney D'Mello, Amy Witherspoon, and Art Graesser. Emote aloud during learning with autotutor: Applying the facial action coding system to cognitive–affective states during learning. *Cognition and Emotion*, 22(5):777–788, 2008.

[CGCB14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[CGSG04] Scotty Craig, Arthur Graesser, Jeremiah Sullins, and Barry Gholson. Affect and learning: an exploratory look into the role of affect in learning with autotutor. *Journal of educational media*, 29(3):241–250, 2004.

[CM] Viviene E. Cree and Cathlin Macaulay, editors. *Transfer of Learning in Professional & Vocational Education*.

[CRK15] Devendra Singh Chaplot, Eunhee Rhim, and Jihie Kim. Predicting student attrition in moocs using sentiment analysis and neural networks. In *AIED Workshops*, 2015.

[Csi90] M Csikszentmihalyi. Flow. the psychology of optimal experience. new york (harperperennial) 1990. 1990.

[CVMBB14] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[dBCA08] Ryan SJ d Baker, Albert T Corbett, and Vincent Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International conference on intelligent tutoring systems*, pages 406–415. Springer, 2008.

[dBCG+10] Ryan S. J. d. Baker, Albert T. Corbett, Sujith M. Gowda, Angela Z. Wagner, Benjamin A. MacLaren, Linda R. Kauffman, Aaron P. Mitchell, and Stephen Giguere. Contextual slip and prediction of student performance after use of an intelligent tutor. In *User Modeling, Adaptation, and Personalization*, pages 52–63. Springer Berlin Heidelberg, 2010.

[dBGW+12] Ryan SJ d Baker, Sujith M Gowda, Michael Wixon, Jessica Kalka, Angela Z Wagner, Aatish Salvi, Vincent Aleven, Gail W Kusbit, Jaclyn Ocumpaugh, and Lisa Rossi. Towards sensor-free affect detection in cognitive tutor algebra. *International Educational Data Mining Society*, 2012.

[DC97] Rob A Dunne and Norm A Campbell. On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*, volume 181, page 185. Citeseer, 1997.

[DCW+08] Sidney K D'mello, Scotty D Craig, Amy Witherspoon, Bethany Mcdaniel, and Arthur Graesser. Automatic detection of learner's affect from conversational cues. *User modeling and user-adapted interaction*, 18(1-2):45–80, 2008.

[DG10] Sidney D'Mello and Art Graesser. Modeling cognitive-affective dynamics with hidden markov models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32, 2010.

[DG11] Sidney D'Mello and Art Graesser. The half-life of cognitive-affective states during complex learning. *Cognition & Emotion*, 25(7):1299–1308, 2011.

[DG12] Sidney D'Mello and Art Graesser. Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2):145–157, 2012.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[DLS+10] Sidney D'Mello, Blair Lehman, Jeremiah Sullins, Rosaire Daigle, Rebekah Combs, Kimberly Vogt, Lydia Perkins, and Art Graesser. A time for emoting: When affect-sensitivity is and isn't effective at promoting deep learning. In *International conference on intelligent tutoring systems*, pages 245–254. Springer, 2010.

[DPMK07] Angela L Duckworth, Christopher Peterson, Michael D Matthews, and Dennis R Kelly. Grit: perseverance and passion for long-term goals. *Journal of personality and social psychology*, 92(6):1087, 2007.

[DSR+15] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, et al. Lasagne: first release. *Zenodo: Geneva, Switzerland*, 3, 2015.

[DTG12] Sidney D'Mello, R.S. Taylor, and A. Graesser. Monitoring affective trajectories during complex learning. In *29th Annual Meeting of the Cognitive Science Society*, pages 203–208. Springer, 2012.

[DWC14] Carol S Dweck, Gregory M Walton, and Geoffrey L Cohen. Academic tenacity: Mindsets and skills that promote long-term learning. *Bill & Melinda Gates Foundation*, 2014.

[EBCV09] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[EJJ04] Andrew Estabrooks, Taeho Jo, and Nathalie Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational intelligence*, 20(1):18–36, 2004.

[EK01] Tapio Elomaa and Matti Kaariainen. An analysis of reduced error pruning. *Journal of Artificial Intelligence Research*, 15:163–187, 2001.

[EM73] Bradley Efron and Carl Morris. Stein's estimation rule and its competitorsan empirical bayes approach. *Journal of the American Statistical Association*, 68(341):117–130, 1973.

[FRA+12] Camille A Farrington, Melissa Roderick, Elaine Allensworth, Jenny Nagaoka, Tasha Seneca Keyes, David W Johnson, and Nicole O Beechum. *Teaching Adolescents to Become Learners: The Role of Noncognitive Factors in Shaping School Performance–A Critical Literature Review.* ERIC, 2012.

[Fyf16] Emily R Fyfe. Providing feedback on computer-based algebra homework in middle-school classrooms. *Computers in Human Behavior*, 63:568–574, 2016.

[Gal86] Francis Galton. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263, 1886.

[GB15] Yue Gong and Joseph E Beck. Towards detecting wheel-spinning: Future failure in mastery learning. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 67–74. ACM, 2015.

[GRD+13] Thea Faye G Guia, MA Mercedes T Rodrigo, Michelle Dagami, C Marie, Jessica O Sugay, Francis Jan P Macam, and Antonija Mitrovic. An exploratory study of factors indicative of affective states of students using sql-tutor. *Research & Practice in Technology Enhanced Learning*, 8(3), 2013.

[GSR+11] Thea Faye G Guia, Jessica O Sugay, Ma Mercedes T Rodrigo, Francis Jan P Macam, Michelle Marie C Dagami, and Antonija Mitrovic. Transitions of affective states in an intelligent tutoring system. *Proceedings of the Philippine Computing Society*, pages 31–35, 2011.

[HH14] Neil T Heffernan and Cristina Lindquist Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[HHSK00] James Heckman, Neil Hohmann, Jeffrey Smith, and Michael Khoo. Substitution and dropout bias in social experiments: A study of an influential social experiment. *The Quarterly Journal of Economics*, 115(2):651–694, 2000.

[Hor65] John L Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185, 1965.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HT01] David J Hand and Robert J Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45(2):171–186, 2001.

[HWBA18] A Harrison, N. Wixon, A. F. Botelho, and I. Arroyo. Sensor-free predictive models of affect in an online learning environment. In *Proceedings of the 11th International Conference on Educational Data Mining*, pages 634–637, 2018.

[Ioa05] John PA Ioannidis. Contradicted and initially stronger effects in highly cited clinical research. *Jama*, 294(2):218–228, 2005.

[JBB+18] Yang Jiang, Nigel Bosch, Ryan S Baker, Luc Paquette, Jaclyn Ocumpaugh, Juliana Ma Alexandra L Andres, Allison L Moore, and Gautam Biswas. Expert feature-engineering vs. deep neural networks: Which is better for sensor-free affect detection? In *International Conference on Artificial Intelligence in Education*, pages 198–211. Springer, 2018.

[Jos05] E Joseph. Engagement tracing: using response times to model student disengagement. *Artificial intelligence in education: Supporting learning through intelligent and socially informed technology*, 125:88, 2005.

[JW06] Jeffrey Johns and Beverly Woolf. A dynamic mixture model to detect student motivation and proficiency. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 163. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[Kap08] Manu Kapur. Productive failure. *Cognition and instruction*, 26(3):379–424, 2008.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KH15] René F Kizilcec and Sherif Halawa. Attrition and achievement gaps in online learning. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 57–66. ACM, 2015.

[KKG16] Tanja Käser, Severin Klingler, and Markus Gross. When to stop?: towards universal instructional policies. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 289–298. ACM, 2016.

[KLM16] Mohammad Khajah, Robert V Lindsey, and Michael C Mozer. How deep is knowledge tracing? In *Proceedings of the 9th International Conference on Educational Data Mining*, pages 94–101, 2016.

[KM16] Kenneth R Koedinger and Elizabeth A McLaughlin. Closing the loop with quantitative cognitive task analysis. *International Educational Data Mining Society*, 2016.

[KRP01] Barry Kort, Rob Reilly, and Rosalind W Picard. An affective model of interplay between emotions and learning: Reengineering educational pedagogy-building a learning companion. In *Advanced Learning Technologies, 2001. Proceedings. IEEE International Conference on*, pages 43–46. IEEE, 2001.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[KVG18a] Byung-Hak Kim, Ethan Vizitei, and Varun Ganapathi. Gritnet 2: Real-time student performance prediction with domain adaptation. *arXiv preprint arXiv:1809.06686*, 2018.

[KVG18b] Byung-Hak Kim, Ethan Vizitei, and Varun Ganapathi. Gritnet: Student performance prediction with deep learning. *arXiv preprint arXiv:1804.07405*, 2018.

[LDG12] Blair Lehman, Sidney D'Mello, and Art Graesser. Confusion and complex learning during interactions with computer learning environments. *The Internet and Higher Education*, 15(3):184–194, 2012.

[Lev66] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[LHOW15] Charles Lang, Neil Heffernan, Korinn Ostrow, and Yutao Wang. The impact of incorporating student confidence items into an intelligent tutor: A randomized controlled trial. *International Educational Data Mining Society*, 2015.

[LMDP08] Blair Lehman, Melanie Matthews, Sidney DMello, and Natalie Person. What are you feeling? investigating student affective states during expert human tutoring sessions. In *International Conference on Intelligent Tutoring Systems*, pages 50–59. Springer, 2008.

[LPOB13] Zhongxiu Liu, Visit Pataranutaporn, Jaclyn Ocumpaugh, and Ryan Baker. Sequences of frustration and confusion, and learning. In *Educational Data Mining 2013*. Citeseer, 2013.

[LSHR15] Anne Lamb, Jascha Smilack, Andrew Ho, and Justin Reich. Addressing common analytic challenges to randomized experiments in moocs: Attrition and zero-inflation. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 21–30. ACM, 2015.

[MCS16] Noboru Matsuda, Sanjay Chandrasekaran, and John C Stamper. How quickly can wheel spinning be detected? In *EDM*, pages 607–608, 2016.

[Mis96] Marianne Miserandino. Children who do well in school: Individual differences in perceived competence and autonomy in above-average children. *Journal of educational psychology*, 88(2):203, 1996.

[MM94] GJ McLachlan and DC McGiffin. On the role of finite mixture models in survival analysis. *Statistical Methods in Medical Research*, 3(3):211–226, 1994. PMID: 7820292.

[MM18] J Makhlouf and T Mine. Predicting if students will pursue a stem career using school-aggregated data from their usage of an intelligent tutoring system. In *Educational Data Mining 2018*, pages 533–536, 2018.

[MMCS11] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011.

[MRS02] Christoph Müller, Stefan Rapp, and Michael Strube. Applying co-training to reference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

[MTL+17] Patrick McGuire, Shihfen Tu, Mary Ellin Logue, Craig A Mason, and Korinn Ostrow. Counterintuitive effects of online feedback in middle school math: results from a randomized controlled trial in assistments. *Educational Media International*, 54(3):231–244, 2017.

[Mur] K. Murphy. Bayes net toolbox for matlab. `https://github.com/bayesnet/bnt/`.

[MW09] W Masten and MO Wright. Resilience over the lifespan. *Handbook of adult resilience*, pages 213–237, 2009.

[MWP+16] Wookhee Min, Joseph B Wiggins, Lydia G Pezzullo, Alexandria K Vail, Kristy Elizabeth Boyer, Bradford W Mott, Megan H Frankosky, Eric N Wiebe, and James C Lester. Predicting dialogue acts for intelligent virtual agents with multimodal student interaction data. *International Educational Data Mining Society*, 2016.

[NJS17] Shinichi Nakagawa, Paul CD Johnson, and Holger Schielzeth. The coefficient of determination r 2 and intra-class correlation coefficient from generalized linear mixed-effects models revisited and expanded. *Journal of the Royal Society Interface*, 14(134):20170213, 2017.

[OAB+17] Jaclyn Ocumpaugh, Juan Miguel Andres, Ryan Baker, Jeanine DeFalco, Luc Paquette, Jonathan Rowe, Bradford Mott, James Lester, Vasiliki Georgoulas, Keith Brawner, et al. Affect dynamics in military trainees using vmedic: From engaged concentration to boredom to confusion. In *International Conference on Artificial Intelligence in Education*, pages 238–249. Springer, 2017.

[OBG+14] Jaclyn Ocumpaugh, Ryan Baker, Sujith Gowda, Neil Heffernan, and Cristina Heffernan. Population validity for educational data mining models: A case study in affect detection. *British Journal of Educational Technology*, 45(3):487–501, 2014.

[OBR15] Jaclyn Ocumpaugh, Ryan Baker, and Ma. Mercedes T. Rodrigo. Baker rodrigo ocumpaugh monitoring protocol (bromp) 2.0 technical and training manual. Technical report, Technical Report. New York, NY:

Teachers College, Columbia University. Manila, Philippines: Ateneo Laboratory for the Learning Sciences, 2015.

[OSW+16] Korinn S Ostrow, Doug Selent, Yan Wang, Eric G Van Inwegen, Neil T Heffernan, and Joseph Jay Williams. The assessment of learning infrastructure (ali): the theory, practice, and scalability of automated assessment. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 279–288. ACM, 2016.

[PBBH13] Maria Ofelia Pedro, Ryan Baker, Alex Bowers, and Neil Heffernan. Predicting college enrollment from student interaction with an intelligent tutoring system in middle school. In *Educational Data Mining 2013*, 2013.

[PBdCO15] Luc Paquette, Ryan S Baker, Adriana de Carvalho, and Jaclyn Ocumpaugh. Cross-system transfer of machine learned and knowledge engineered models of gaming the system. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 183–194. Springer, 2015.

[PBH+15] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513, 2015.

[PBSP+13] Zachary A Pardos, Ryan SJD Baker, Maria OCZ San Pedro, Sujith M Gowda, and Supreeth M Gowda. Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 117–124. ACM, 2013.

[PBSP+14] Zachary A Pardos, RS Baker, MOCZ San Pedro, Sujith M Gowda, and Supreeth M Gowda. Affective states and state tests: investigating how affect and engagement during the school year predict end-of-year learning outcomes. *Journal of Learning Analytics*, 1(1):107–128, 2014.

[PdCBO14] Luc Paquette, Adriana de Carvahlo, Ryan Baker, and Jaclyn Ocumpaugh. Reengineering the feature distillation process: A case study in detection of gaming the system. In *Educational data mining 2014*. Citeseer, 2014.

[PGMK14] John F Pane, Beth Ann Griffin, Daniel F McCaffrey, and Rita Karam. Effectiveness of cognitive tutor algebra i at scale. *Educational Evaluation and Policy Analysis*, 36(2):127–144, 2014.

[PH10a] Zachary A Pardos and Neil T Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.

[PH10b] Zachary A Pardos and Neil T Heffernan. Navigating the parameter space of bayesian knowledge tracing models: Visualizations of the convergence of the expectation maximization algorithm. *EDM*, 2010:161–170, 2010.

[PH11] Zachary A Pardos and Neil T Heffernan. Kt-idem: introducing item difficulty to the knowledge tracing model. In *International conference on user modeling, adaptation, and personalization*, pages 243–254. Springer, 2011.

[PHM+18] Zachary A Pardos, Changran Hu, Pengqiu Meng, Michael Neff, and Dor Abrahamson. Classifying learner behavior from high frequency touchscreen data using recurrent neural networks. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*, pages 317–322. ACM, 2018.

[PJCK09] Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*, 2009.

[Pra93] Lorien Y Pratt. Discriminability-based transfer between neural networks. In *Advances in neural information processing systems*, pages 204–211, 1993.

[PRB+16] Luc Paquette, Jonathan Rowe, Ryan Baker, Bradford Mott, James Lester, Jeanine DeFalco, Keith Brawner, Robert Sottilare, and Vasiliki Georgoulas. Sensor-free or sensor-full: A comparison of data modalities in multi-channel affect detection. *International Educational Data Mining Society*, 2016.

[PS+04] Christopher Peterson, Martin EP Seligman, et al. *Character strengths and virtues: A handbook and classification*, volume 1. Oxford University Press, 2004.

[PSB+17] Thanaporn Patikorn, Douglas Selent, J Beck, N Heffernan, and J Zhou. Using a single model trained across multiple experiments to improve the detection of treatment effects. In *10th International Conference on Educational Data Mining*, 2017.

[PSC93] Brian C Patrick, Ellen A Skinner, and James P Connell. What motivates children's behavior and emotion? joint effects of perceived control and autonomy in the academic domain. *Journal of Personality and social Psychology*, 65(4):781, 1993.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RBA+11] Ma Mercedes T Rodrigo, RSJ d Baker, J Agapito, J Nabos, MC Repalam, SS Reyes Jr, and MOC San Pedro. The effects of an embodied conversational agent on student affective dynamics while using an intelligent tutoring system. *IEEE Transactions on Affective Computing*, 2(4):18–37, 2011.

[RBA+12] Ma Mercedes T Rodrigo, Ryan SJd Baker, Jenilyn Agapito, Julieta Nabos, Ma Concepcion Repalam, Salvador S Reyes, and Maria Ofelia CZ San Pedro. The effects of an interactive software agent on student affective dynamics while using; an intelligent tutoring system. *IEEE Transactions on Affective Computing*, 3(2):224–236, 2012.

[RBJ+09] Ma Mercedes T Rodrigo, Ryan S Baker, Matthew C Jadud, Anna Christine M Amarra, Thomas Dy, Maria Beatriz V Espejo-Lahoz, Sheryl Ann L Lim, Sheila AMS Pascua, Jessica O Sugay, and Emily S Tabanao. Affective and behavioral predictors of novice programmer achievement. In *ACM SIGCSE Bulletin*, volume 41, pages 156–160. ACM, 2009.

[RC07] Rod D Roscoe and Michelene TH Chi. Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors explanations and questions. *Review of Educational Research*, 77(4):534–574, 2007.

[RCY+14] Carolyn Penstein Rosé, Ryan Carlson, Diyi Yang, Miaomiao Wen, Lauren Resnick, Pam Goldman, and Jennifer Sherer. Social factors that contribute to attrition in moocs. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 197–198. ACM, 2014.

[Red16] Gloria Nogueiras Redondo. Turning points en las trayectorias emocionales de estudiantes en un contexto desafiante de aprendizaje experiencial: una aproximación dinámica. In *Quintas Jornadas de Jóvenes Investigadores de la Universidad de Alcalá: Humanidades y Ciencias Sociales*, pages 245–254. Servicio de Publicaciones, 2016.

[RFMM16] Jeremy Roschelle, Mingyu Feng, Robert F Murphy, and Craig A Mason. Online mathematics homework increases student achievement. *AERA Open*, 2(4):2332858416673968, 2016.

[RRMB+08] Ma Mercedes T Rodrigo, Genaro Rebolledo-Mendez, RSJd Baker, Benedict du Boulay, JO Sugay, SAL Lim, MB Espejo-Lahoz, and R Luckin. The effects of motivational modeling on affect in an intelligent tutoring system. In *Proceedings of International Conference on Computers in Education*, volume 57, page 64, 2008.

[RTH+11] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC bioinformatics*, 12(1):77, 2011.

[Rub81] Donald B Rubin. Estimation in parallel randomized experiments. *Journal of Educational Statistics*, 6(4):377–401, 1981.

[S+90] Stephen M Stigler et al. The 1988 neyman memorial lecture: a galtonian perspective on shrinkage estimators. *Statistical Science*, 5(1):147–155, 1990.

[SBO+16] Stefan Slater, Ryan Baker, Jaclyn Ocumpaugh, Paul Inventado, Peter Scupelli, and Neil Heffernan. Semantic features of math problems: Relationships to student learning and engagement. *International Educational Data Mining Society*, 2016.

[SBPH18] Adam C Sales, Anthony Botelho, Thanaporn Patikorn, and Neil T Heffernan. Using big data to sharpen design-based inference in a/b tests. In *Proceedings of the 11th International Conference on Educational Data Mining*, pages 479–486,, 2018.

[SC74] Richard L Solomon and John D Corbit. An opponent-process theory of motivation: I. temporal dynamics of affect. *Psychological review*, 81(2):119, 1974.

[SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[SHR18] Adam C Sales, Ben B Hansen, and Brian Rowan. Rebar: Reinforcing a matching estimator with predictions from high-dimensional covariates. *Journal of Educational and Behavioral Statistics*, 43(1):3–31, 2018.

[SLMN11] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 129–136, 2011.

[SML11] Jennifer Sabourin, Bradford Mott, and James C Lester. Modeling learner affect with theoretically grounded dynamic bayesian networks. In *International Conference on Affective Computing and Intelligent Interaction*, pages 286–295. Springer, 2011.

[SMSB14] Sergio Salmeron-Majadas, Olga C Santos, and Jesus G Boticario. An evaluation of mouse and keyboard interaction indicators towards non-intrusive and low cost affective modeling in an educational context. *Procedia Computer Science*, 35:691–700, 2014.

[SPdBR11] Maria Ofelia Clarissa Z San Pedro, Ryan SJ d Baker, and Ma Mercedes T Rodrigo. Detecting carelessness through contextual estimation of slip probabilities among students using an intelligent tutor for mathematics. In *International Conference on Artificial Intelligence in Education*, pages 304–311. Springer, 2011.

[SPH16] Douglas Selent, Thanaporn Patikorn, and Neil Heffernan. Assistments dataset from multiple randomized controlled experiments. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*, pages 181–184. ACM, 2016.

[SPOBH14] Maria Ofelia San Pedro, Jaclyn Ocumpaugh, Ryan S Baker, and Neil T Heffernan. Predicting stem and non-stem college major enrollment from middle school interaction with mathematics educational software. In *Educational Data Mining 2014*, pages 276–279, 2014.

[SS14] Mike Sharkey and Robert Sanders. A process for predicting mooc attrition. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 50–54, 2014.

[TARA+16] The Theano Development Team, Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, et al. Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

[WA18] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

[War15] Hiroko Kawaguchi Warshauer. Productive struggle in middle school mathematics classrooms. *Journal of Mathematics Teacher Education*, 18(4):375–400, 2015.

[WGM06] Christopher O Walker, Barbara A Greene, and Robert A Mansell. Identification with academics, intrinsic/extrinsic motivation, and self-efficacy as predictors of cognitive engagement. *Learning and individual differences*, 16(1):1–12, 2006.

[WH13] Yutao Wang and Neil Heffernan. Extending knowledge tracing to allow partial credit: Using continuous versus binary nodes. In *International conference on artificial intelligence in education*, pages 181–188. Springer, 2013.

[WHH15] Yutao Wang, Neil T Heffernan, and Cristina Heffernan. Towards better affect detectors: effect of missing skills, class features and common wrong answers. In *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge*, pages 31–35. ACM, 2015.

[Wil27] Edwin B Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

[WJ09] Pedro A Willging and Scott D Johnson. Factors that influence students' decision to dropout of online courses. *Journal of Asynchronous Learning Networks*, 13(3):115–127, 2009.

[WZ89] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[XBL13] Xiaolu Xiong, JosephE. Beck, and Shoujing Li. Class distinctions: Leveraging class-level features to predict student retention performance. In H.Chad Lane, Kalina Yacef, Jack Mostow, and Philip Pavlik, editors, *Artificial Intelligence in Education*, volume 7926 of *Lecture Notes in Computer Science*, pages 820–823. Springer Berlin Heidelberg, 2013.

[XCSM16] Wanli Xing, Xin Chen, Jared Stein, and Michael Marcinkowski. Temporal predication of dropouts in moocs: Reaching the low hanging fruit through stacking generalization. *Computers in Human Behavior*, 58:119–129, 2016.

[XLB13] Xiaolu Xiong, Shoujing Li, and Joseph E Beck. Will you get it right next week: Predict delayed performance in enhanced its mastery cycle. In *FLAIRS Conference*, 2013.

[XZVIB16] Xiaolu Xiong, Siyuan Zhao, Eric Van Inwegen, and Joseph Beck. Going deeper with deep knowledge tracing. In *Proceedings of the 9th International Conference on Educational Data Mining*, pages 545–550, 2016.

[YLYY18] Chun-kit Yeung, Zizheng Lin, Kai Yang, and Dit-yan Yeung. Incorporating features learned by an enhanced deep knowledge tracing model for stem/non-stem job prediction. *arXiv preprint arXiv:1806.03256*, 2018.

[YSAR13] Diyi Yang, Tanmay Sinha, David Adamson, and Carolyn Penstein Rosé. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-driven education workshop*, volume 11, page 14, 2013.

[YY18] Chun-Kit Yeung and Dit-Yan Yeung. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. *arXiv preprint arXiv:1806.02180*, 2018.

[ZXZ+17] Liang Zhang, Xiaolu Xiong, Siyuan Zhao, Anthony Botelho, and Neil T Heffernan. Incorporating rich features into deep knowledge tracing. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 169–172. ACM, 2017.

Chapter 7: DRIVER-SEAT References and Supplemental Figures

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, *4*(2), 167-207.

Aung, A.M., & Whitehill, J. (2018a). Harnessing Label Uncertainty to Improve Modeling: An Application to Student Engagement Recognition. Under review.

Aung, A.M., & Whitehill, J. (2018b). Automatic Eye-Gaze Following for Classroom Observation Analysis. Work in progress. https://users.wpi.edu/~jrwhitehill/AungWhitehill_EyeGazeFollowig_TechReport_Jan2018.pdf

Beck, J. E., & Gong, Y. (2013). Wheel-spinning: Students who fail to master a skill. In *International Conference on Artificial Intelligence in Education* (pp. 431-440). Springer, Berlin, Heidelberg.

Bill & Melinda Gates Foundation. (2015). Teachers know best: Making data work for teachers and students.

Botelho, A. F., Baker, R. S., & Heffernan, N. T. (2017). Improving sensor-free affect detection using deep learning. In E. Andre' et al (Eds.) *Proceedings of the Eighteenth International Conference on Artificial Intelligence in Education, 40-51.*

Botelho, A. F., Ostrow, K. S., Heffernan, N. T. (in submission). ObserveME: Development and application of an observational coding protocol for mental effort. Manuscript submitted for publication.

Craig, S., Graesser, A., Sullins, J., & Gholson, B. (2004). Affect and learning: an exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media, 29*(3), 241-250.

Csikszentmihalyi, M. (1990). Flow: the psychology of optimal experience. New York: Harper-Row.

Diaz, N., & Jones, C. (2017) Cloud based infrastructure for educational deep learning. A WPI Undergraduate senior honors thesis document. Retrieved from the WPI library at https://web.wpi.edu/Pubs/E-project/Available/E-project-042517-224031/unrestricted/ndiaz_ctjones_mqp_paper.pdf

Graesser, A., Chipman, P., Haynes, B., & Olney, A. (2005). AutoTutor: an intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education 48*(4). Retrieved from tranhttp://ieeexplore.ieee.org/document/1532370/.

Heffernan, N. & Heffernan, C. (2014). The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*. 24(4), 470-497. Link to the Springer version DOI 10.1007/s40593-014-0024-x. The Special Issue focused on landmark systems. Retrieved from https://link.springer.com/article/10.1007/s40593-014-0024-x

Heffernan, N, & Heffernan, C. (2016) The Heffernans were invited to speak at the OSTP White House event. Retrieved from http://www.aboutus.assistments.org/homework-immediate-feedback---1-year-study.php

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Hughes, J. N., Cavell, T. A., & Willson, V. (2001). Further support for the developmental significance of the quality of the teacher–student relationship. *Journal of School Psychology, 39*(4), 289-301.

IMS. (2017). Caliper Analytics. Retrieved from http://www.imsglobal.org/activity/caliper.

Johansson, F., Shalit, U. & Sontag, D. (2016). Learning representations for counterfactual inference. In *Proceedings of The 33rd International Conference on Machine Learning*, 3020–3029.

Jiang, H., Dykstra, K., & Whitehill, J. (2018). Predicting When Teachers Look at Their Students in 1-on-1 Tutoring Sessions. Under review.

Kai, S., Almeda, M., Baker, R., Heffernan, C., Heffernan, N. (in press). Modeling wheel-spinning and productive persistence in SkillBuilders. To appear in *Journal of Educational Data Mining*.

Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., Corrado, G., Kukacs, L., Ganea, M., Young, P & Ramavajjala, V. (2016). Smart Reply: Automated response suggestion for email. *KDD '16 August 13-17*, 2016, San Francisco, CA, USA. Retrieved from https://dl.acm.org/citation.cfm?id=2939801.

Kelly, K., Heffernan, N., Heffernan, C., Goldman, S., Pellegrino, G. & Soffer, D. (2013). Estimating the effect of web-based homework. In Lane, Yacef, Mostow & Pavlik (Eds) *The Artificial Intelligence in Education Conference.* pp. 824-827. (Watch the videos and related archived material here: http://web.cs.wpi.edu/~nth/PublicScienceArchive/Kelly.htm and permanently archived at http://www.webcitation.org/6E6lv54G8)

Khajah, M., Lindsey, R. & Mozer, M. (2016). How deep is knowledge tracing? In T. Barnes, M. Chi, and M. Feng, editors, *Proceedings of the 9th International Conference on Educational Data Mining (EDM 2016),* 94–101.

Kort, B., Reilly, R. & Picard, R. (2001). An affective model of interplay between emotions and learning: reengineering educational pedagogy—building a learning companion. Proceedings *IEEE International Conference on Advanced Learning Technology: Issues, Achievements and Challenges*. Madison, Wisconsin: IEEE Computer Society, 43–48.

Littlewort, G., Whitehill, J., Wu, T., Fasel, I., Frank, M., Movellan, J., & Bartlett, M. (2011). The computer expression recognition toolbox (CERT). In *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*, 298-305.

Long, M., Zhu, H., Wang, J., & Jordan, M. I. (2016). Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems 29*, 136–144. Curran Associates, Inc.

Lu, J. (2008). Effects of traditional and digital media on student learning in space design. *The Scholarship of Teaching and Learning at EMU, 2*(5), 75-90.

Miserandino, M. (1996). Children who do well in school: individual differences in perceived competence and autonomy in above-average children. *Journal of Educational Psychology, 88*(2), 203–214.

Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., & Heffernan, C. (2014). Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology, 45*(3), 487-501.

Ocumpaugh, J., Baker, R., & Rodrigo, M.M.T. (2015). Baker Rodrigo Ocumpaugh monitoring protocol (BROMP) 2.0 technical and training manual. New York, NY and Manila, Philippines: Teachers College, Columbia University and Ateneo Laboratory for the Learning Sciences.

Pardos, Z. A., Baker, R. S., San Pedro, M., Gowda, S. M., & Gowda, S. M. (2014). Affective states and state tests: investigating how affect and engagement during the school year predict end-of-year learning outcomes. *Journal of Learning Analytics, 1*(1), 107-128.

Pardos, Z., Trivedi, S., Heffernan, N. & Sarkozy, G. (2012). Clustered knowledge tracing. *11th International Conference on Intelligent Tutoring Systems*. 404-410.

Patrick, B. C., Skinner, E. A. & Connell, J. P. (1993). What motivates children's behavior and emotion? Joint effects of perceived control and autonomy in the academic domain. *Journal of Personality and Social Psychology*, *65*(4), 781–791.

Pavlik Jr., P. I., Cen, H., & Koedinger, K. R. (2009). Performance factors analysis--A new alternative to knowledge tracing. *Online Submission*. https://dl.acm.org/citation.cfm?id=1659529. In *Proceedings of the 2009 conference on Artificial Intelligence in Education*. IOS Press Amsterdam, The Netherlands, 531-538.

Pawlowski, J.M. & Bick, M. (2012). Open educational resources. *Business and Information Systems Engineering*. *4*(4), 209-212.

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, 505-513.

Radius Global Market Research (2017). Massachusetts K-12 education degree power scores: Preference for teachers graduating from leading institutions. White Paper.

RAND Education. (2012). Teachers Matter: Understanding teachers' impact on student achievement. Santa Monica, CA: https://www.rand.org/pubs/corporate_pubs/CP693z1-2012-09.html.

Rimm-Kaufman, S. E., Baroody, A., Larsen, R., Curby, T. W., & Abry, T. (2015). To what extent do teacher-student interaction quality and student gender contribute to fifth graders' engagement in

mathematics learning? *Journal of Educational Psychology*, 107(1), 170-185. doi: http://dx.doi.org/10.1037/a0037252

Roschelle, J., Feng, M., Murphy, R.F., & Mason, C.A. (2016). Online mathematics homework increases student achievement. *AERA Open, 2(4)*: 1-12. Retreived from http://journals.sagepub.com/doi/abs/10.1177/2332858416673968

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation (No. ICS-8506). California Univ San Diego La Jolla Inst for Cognitive Science.

San Pedro, M.O.Z., Baker, R.S.J.d., Bowers, A.J., Heffernan, N.T. (2013) Predicting College Enrollment from Student Interaction with an Intelligent Tutoring System in Middle School. *Proceedings of the 6th International Conference on Educational Data Mining*, 177-184.

San Pedro, M., Ocumpaugh, J., Baker, R., & Heffernan, N. (2014) Predicting STEM and Non-STEM College Major Enrollment from Middle School Interaction with Mathematics Educational Software. In John Stamper et al. (Eds) *Proceedings of the 7th International Conference on Educational Data Mining*. pp. 276-279.

Singer, N. (2017, May 17) How GOOGLE took over the classroom. The New York Time. Retrieved from https://www.nytimes.com/2017/05/13/technology/google-education-chromebooks-schools.html?_r=0 May 17, 2017.

Song, F., Trivedi, S., Wang, Y., Sárközy, G, & Heffernan, N. (2013). Applying Clustering to the Problem of Predicting Retention within an ITS: Comparing Regularity Clustering with Traditional Methods. In Boonthum-Denecke, Youngblood (Eds) *Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2013*, St. Pete Beach, Florida. May 22-24, 2013. AAAI Press, 527-532.

Steenbergen-Hu, S. & Cooper, H. (2013). A meta-analysis of the effectiveness of intelligent tutoring systems on K–12 students' mathematical learning. *Journal of Educational Psychology, 105*(4), Nov 2013, 970-987. Retrieved from http://psycnet.apa.org/record/2013-31551-001

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems (NIPS)*.

Tran, K., Bisazza, A., & Monz, C. (2016). Recurrent memory network for language modeling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL HLT)*.

Trivedi, S., Pardos, Z. & Heffernan, N. (2011). Clustering Students to Generate an Ensemble to Improve Standard Test Score Predictions In Biswas et al. (Eds) *Proceedings of the Artificial Intelligence in Education Conference 2011,* 328–336.

Trivedi, S. Pardos, *Z.,* Sarkozy, G. & Heffernan, N. (2012). Co-Clustering by Bipartite Spectral Graph Partitioning for Out-Of-Tutor Prediction. *5th International Conference on Educational Data Mining*. pp. 33-40

Vinyals, O. & Le, Q. V. (2015). A neural conversation model. In *International Conference on Machine Learning Deep Learning Workshop*.

Wang, Y., Heffernan, N. T., & Heffernan, C. (2015). Towards better affect detectors: effect of missing skills, class features and common wrong answers. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, 31-35.

Whitehill, J., Littlewort, G., Fasel, I., Bartlett, M., & Movellan, J. (2009). Toward practical smile detection. *IEEE transactions on pattern analysis and machine intelligence. 31*(11), 2106-2111.

Whitehill, J., Mohan, K., Seaton, D., Rosen, Y., & Tingley, D. (2017). MOOC Dropout Prediction: How to Measure Accuracy?. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, 161-164.

Whitehill, J., Serpell, Z., Lin, Y. C., Foster, A., & Movellan, J. R. (2014). The faces of engagement: Automatic recognition of student engagement from facial expressions. *IEEE Transactions on Affective Computing, 5*(1), 86-98.

Whitehill, J., Williams, J., Lopez, G., Coleman, C., & Reich, J. (2015). Beyond Prediction: First Steps toward Automatic Intervention in MOOC Student Stopout. *International Educational Data Mining Society*.

Wiliam, D. (1999). Formative assessment in mathematics Part 2: Feedback, *Equals: Mathematics and Special Educational Needs, 5*(3) 8-11.

Williams, J. J., Kim, J., Rafferty, A., Maldonado, S., Gajos, K. Z., Lasecki, W. S. & Heffernan, N. T. (2016) Axis: Generating explanations at scale with learnsourcing and machine learning *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*, 379-388.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation, 1*(2), 270-280.

Wilson, K., Xiong, X., Khajah, M., Lindsey, R. V., Zhao, S., Karklin, K., Van Inwegen, E., Han, B., Ekanadham, C., Beck, J., Heffernan, N., & Mozer, M., (2016) Estimating student proficiency: Deep learning is not the panacea. Submission to the *NIPS 2016 Workshop on Machine Learning for Education.*

xAPI. (2017). What is the Experience API? Retrieved from https://experienceapi.com/overview.

Xiong, X., Zhao, S., Van Inwegen, E., & Beck, J. (2016). Going deeper with deep knowledge tracing. In *Proceedings of the 7th International Conference on Educational Data Mining*, 545-550.

Zhang, L., Xiong, X., Zhao, S., Botelho, A. & Heffernan, N. (2017) Incorporating Rich Features into Deep Knowledge Tracing. In the *Proceedings of the Forth (2017) ACM Conference on Learning @ Scale.(L@S2017)* Cambridge, MA, 169-172.

Zhao, S . & Heffernan, N. (2017) Estimating Individual Treatment Effects from Educational Studies with Residual Counterfactual Networks. In *Proceedings of the 10th International Conference on Educational Data Mining (EDM 2017)*.

Zhao, S., Zhang, Y., Xiong, X., Botelho, A. F., & Heffernan, N. T. (2017) A Memory-Augmented Neural Model for Automated Grading. In the *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale. (L@S2017)* Cambridge, MA, 189-172.

References from Prior NSF Funding.
According to: http://www.nsf.gov/pubs/policydocs/pappguide/nsf14001/gpg_2.jsp, respondents must include "the publications resulting from the NSF award" and "a complete bibliographic citation for each publication must be provided either in this section or in the References Cited section of the proposal". Therefore, below are the references that resulted from prior NSF support (separated by each grant & CoPI for each CoPI that actually has Prior NSF Funding).

Results of Prior NSF Funding for Heffernan's "CAREER- Learning about Learning" grant (#0448319 $600,000).

1. Bahador, N., Pardos, Z., Heffernan & Baker, R. (2011). Less is More: Improving the Speed and Prediction Power of Knowledge Tracing by Using Less Data In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., and Stamper, J. (Eds.) *Proceedings of the 4th International Conference on Educational Data Mining.* pp. 101-110.
2. Baker, R., Pardos, Z., Gowda, S., Nooraei, B., & Heffernan, N. (2011). Ensembling Predictions of Student Knowledge within Intelligent Tutoring Systems. In Konstant et al. (Eds.) *20th International Conference on User Modeling, Adaptation and Personalization (UMAP 2011).* pp. 13-24.
3. Baker, R., Walonoski, J., Heffernan, T., Roll, I., Corbett, A. & Koedinger, K. (2008). Why students engage in "Gaming the System" behavior in interactive learning environments. *Journal of Interactive Learning Research (JILR).19*(2), 185-224.
4. Feng, M. & Heffernan, N. (2010). Can We Get Better Assessment From a Tutoring System Compared to Traditional Paper Testing? Can We Have Our Cake (Better Assessment) and Eat It Too (Student Learning During the Test) In Baker, R.S.J.d., Merceron, A., Pavlik, P.I. Jr. (Eds.) *Proceedings of the 3rd International Conference on Educational Data Mining.* pp. 41-50.
5. Feng, M., & Heffernan, N.T. (2006). Informing teachers live about student learning: Reporting in the Assistment system *Technology, Instruction, Cognition, and Learning Journal. 3*(1-2), 63.
6. Feng, M., & Heffernan, N.T. (2007). Towards live informing and automatic analyzing of student learning: Reporting in the Assistment system. *Journal of Interactive Learning Research (JILR) 18*(2), 207-230.
7. Feng, M., Beck, J., & Heffernan, N. (2009). Using Learning Decomposition and Bootstrapping with Randomization to Compare the Impact of Different Educational Interventions on Learning. In Barnes, Desmarais, Romero & Ventura (Eds) *Proc. of the 2nd International Conference on Educational Data Mining.* pp. 51-60.
8. Feng, M., Beck, J,. Heffernan, N. & Koedinger, K. (2008). Can an Intelligent Tutoring System Predict Math Proficiency as Well as a Standardized Test? In Baker & Beck (Eds.). *Proceedings of the 1st International Conference on Education Data Mining*. Montreal, Canada. pp. 107-116.
9. Feng, M., Heffernan, N. & Beck, J. (2009). Using Learning Decomposition to Analyze Instructional Effectiveness in the ASSISTment System. *Proceedings of the 2009 Artificial Intelligence in Education Conference*. IOS Press. pp. 523-530.
10. Feng, M., Heffernan, N., Beck, J, & Koedinger, K. (2008). Can we predict which groups of questions students will learn from? In Baker & Beck (Eds.). *Proceedings of the 1st International Conference on Education Data Mining*. Montreal, Canada. pp. 218-225.
11. Feng, M., Heffernan, N. & Koedinger, K.R. (2006a). Predicting state test scores better with intelligent tutoring systems: developing metrics to measure assistance required. In Ikeda, Ashley & Chan (Eds.). *Proceedings of the Eighth International Conference on Intelligent Tutoring Systems.* Springer-Verlag: Berlin. pp. 31-40.
12. Feng, M., Heffernan, N. T., & Koedinger, K. R. (2006b). Addressing the testing challenge with a Web-based e-assessment system that tutors as it assesses. *Proceedings of the Fifteenth International*

World Wide Web Conference (WWW-06). New York, NY: ACM Press. ISBN:1-59593-332-9. pp. 307-316.
13. Feng, M., Heffernan, N.T., & Koedinger, K.R. (2009). Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. The Journal of *User Modeling and User-Adapted Interaction.19*, 243-266.
14. Feng, M., Heffernan, N. T., Mani, M., & Heffernan, C. (2007). Assessing students' performance longitudinally: Item difficulty parameter vs. skill learning tracking. The National Council on Educational Measurement 2007 Annual Conference, Chicago.
15. Feng, M., Heffernan, N.T., Heffernan, & C., Mani, M. (2009). Using Mixed-Effects Modeling to Analyze Different Grain-Sized Skill Models. IEEE Transactions on Learning Technologies, *2*(2), 79-92.
16. Gong, Y., Beck, J, Heffernan, N. (2010). Using Multiple Dirichlet distributions to improve parameter plausibility Educational Data Mining 2010. In Baker, R.S.J.d., Merceron, A., Pavlik, P.I. Jr. (Eds.) *Proceedings of the 3rd International Conference on Educational Data Mining*. pp. 61-70.
17. Gong, Y, Beck, J. E., Heffernan, N. T. (2011). How to Construct More Accurate Student Models: Comparing and Optimizing Knowledge Tracing and Performance Factor Analysis. *International Journal of Artificial Intelligence in Education*. *21*, 27-46.
18. Gong, Y., Beck, J. & Heffernan, N. (2010). Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting. In Aleven, V., Kay, J & Mostow, J. (Eds) *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010) Part 1*. Springer-Verlag, Berlin. pp. 35-44.
19. Gong, Y., Beck, J. & Heffernan, N. (2012). WEBsistments: Enabling an Intelligent Tutoring System to Excel at Explaining *Why* Other Than Showing *How*; 11th International Conference on Intelligent Tutoring Systems. Springer. pp 268-273
20. Gong, Y., Beck, J., Heffernan, N. & Forbes-Summers, E. (2010). The impact of gaming (?) on learning at the fine-grained level. In Aleven, V., Kay, J & Mostow, J. (Eds) *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010) Part 1*. Springer. pp.194-203.
21. Gong, Y., Rai, D. Beck, J. & Heffernan, N. (2009). Does Self-Discipline impact students' knowledge and learning? In Barnes, Desmarais, Romero & Ventura (Eds) *Proc. of the 2nd International Conference on Educational Data Mining.* pp. 61-70. ISBN: 978-84-613-2308-1.
22. Gowda, S., Baker, R.S.J.d., Pardos, Z., Heffernan, N. (2011). The Sum is Greater than the Parts: Ensembling Student Knowledge Models in ASSISTments. Proceedings of the KDD 2011 Workshop on KDD in Educational Data.
23. Hawkins, W., Baker, R. S. J. d., & Heffernan, N. T., (2013). Which is more responsible for boredom in intelligent tutoring systems: students (trait) or problems (state)? Affective Computing and Intelligent Interaction. Geneva. pp. 618-623.
24. Hawkins, W., Heffernan, N., Wang, Y. & Baker, S,J,d.. (2013). Extending the Assistance Model: Analyzing the Use of Assistance over Time. In S. D'Mello, R. Calvo, & A. Olney (Eds.) Proceedings of the 6th International Conference on Educational Data Mining (EDM2013). Memphis, TN. pp. 59-66.
25. Heffernan, N. & Heffernan, C. (2014). The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. *International Journal of Artificial Intelligence in Education. 24 (4),* 470-497.
26. Heffernan, N. T., Koedinger, K. & Razzaq, L. (2008). Expanding the model-tracing architecture: A 3rd generation intelligent tutor for Algebra symbolization. *The International Journal of Artificial Intelligence in Education. 18*(2), 153-178.
27. Heffernan N.T., Turner T. E., Lourenco A.L.N., Macasek M.A., Nuzzo-Jones G., & Koedinger K.R. (2006). The ASSISTment builder: Towards an analysis of cost effectiveness of ITS

creation.*Proceedings of the 19th International FLAIRS Conference*, Melbourne Beach, Florida, USA. pp. 515-520.
28. Koedinger, K., McLaughlin, E. & Heffernan, N. (2010). A Quasi-Experimental Evaluation of an On-line Formative Assessment and Tutoring System. Journal of Educational Computing Research. Baywood Publishing. *4*, 489 - 510.
29. Militello, M., & Heffernan, N. (2009). Which one is "just right"? What educators should know about formative assessment systems. *International Journal of Educational Leadership Preparation, 4*(3), 1-8.
30. Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., Heffernan, C. (2014). Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology, 45* (3), 487-501.
31. Pardos, Z. & Heffernan, N. (2009). Detecting the Learning Value of Items in a Randomized Problem Set. In Dimitrova, Mizoguchi, du Boulay & Graesser (Eds.) *Proceedings of the 2009 Artificial Intelligence in Education Conference.* IOS Press. pp. 499-506.
32. Pardos, Z. & Heffernan, N. (2010). Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In Paul De Bra, Alfred Kobsa, David Chin, (Eds.) *The 18th Proceedings of the International Conference on User Modeling, Adaptation and Personalization*. pp. 255-266.
33. Pardos, Z. & Heffernan, N. (2010). Navigating the parameter space of Bayesian Knowledge Tracing models: Visualization of the convergence of the Expectation Maximization algorithm. In Baker, R.S.J.d., Merceron, A., Pavlik, P.I. Jr. (Eds.) *Proceedings of the 3rd International Conference on Educational Data Mining.* pp. 161-170.
34. Pardos, Z. & Heffernan, N. (2011). KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In Konstant et al. (Eds.) *20th International Conference on User Modeling, Adaptation and Personalization (UMAP 2011).* pp. 243-254.
35. Pardos, Z. & Heffernan, N. (2012). Tutor Modeling vs. Student Modeling. Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference. Invited talk. Florida Artificial Intelligence Research Society (FLAIRS 2012). St. Peter Beach, Florida pp 420-425.
36. Pardos, Z. A., Beck, J., Ruiz, C. & Heffernan, N. T. (2008). The Composition Effect: Conjunctive or Compensatory? An Analysis of Multi-Skill Math Questions in ITS. In Baker & Beck (Eds.) *Proceedings of the First International Conference on Educational Data Mining.* Montreal, Canada. pp. 147-156.
37. Pardos, Z. A., Heffernan, N. T., Anderson, B. & Heffernan, C. (2007). The effect of model granularity on student performance prediction using Bayesian networks. The International User Modeling Conference 2007. pp. 435-439.
38. Pardos, Z., Gowda, S., Baker, R. & Heffernan, N. (2011). Ensembling Predictions of Student Post-Test Scores for an Intelligent Tutoring System. In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., and Stamper, J. (Eds.) *Proceedings of the 4th International Conference on Educational Data Mining.* pp. 189-198.
39. Pardos, Z., Trivedi, S., Heffernan, N. & Sarkozy, G. (2012). Clustered Knowledge Tracing. 11th International Conference on Intelligent Tutoring Systems. pp 404-410
40. Pardos, Z.A., & Heffernan, N.T. (2009). Determining the Significance of Item Order In Randomized Problem Sets. In Barnes, Desmarais, Romero & Ventura (Eds.) *Proc. of the 2nd International Conference on Educational Data Mining*. pp. 111-120.
41. Pardos, Z.A., Gowda, S. M., Baker, R. S.J.D., Heffernan, N. T., (2012). The Sum is Greater than the Parts: Ensembling Models of Student Knowledge in Educational Software. *ACM's Knowledge Discovery and Datamining Explorations, 13*(2), 37-44

42. Qiu, Y., Pardos, *Z.* & Heffernan, N. (2012). Towards data driven user model improvement. Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference. Florida Artificial Intelligence Research Society (FLAIRS 2012). pp. 462-465.
43. Qiu, Y., Qi, Y., Lu, H., Pardos, Z. & Heffernan, N. (2011). Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., and Stamper, J. (Eds.) *Proceedings of the 4th International Conference on Educational Data Mining.* pp.139-148.
44. Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K. R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar. R, Walonoski, J.A., Macasek. M.A. & Rasmussen, K.P. (2005). The Assistment project: Blending assessment and assisting. In C.K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.) *Proceedings of the 12th Artificial Intelligence in Education,* Amsterdam: ISO Press. pp. 555-562.
45. Razzaq, L., Heffernan, N., Feng, M., & Pardos Z. (2007). Developing Fine-Grained Transfer Models in the ASSISTment System. *Journal of Technology, Instruction, Cognition, and Learning. 5*(3), 289-304.
46. Razzaq, L., Mendicino, M. & Heffernan, N. (2008). Comparing classroom problem-solving with no feedback to web-based homework assistance. In Woolf, Aimeur, Nkambou and Lajoie (Eds.) *Proceeding of the 9th International Conference on Intelligent Tutoring Systems.* pp. 426 -437.
47. Razzaq, L., Patvarczki, J., Almeida, S.F., Vartak, M., Feng, M., Heffernan, N.T. and Koedinger, K. (2009). The ASSISTment builder: Supporting the Life-cycle of ITS Content Creation. IEEE Transactions on Learning Technologies Special Issue on Real-World Applications of Intelligent Tutoring Systems. *2*(2) 157-166.
48. Song, F., Trivedi, S., Wang, Y., Sárközy, G, & Heffernan, N. (2013). Applying Clustering to the Problem of Predicting Retention within an ITS: Comparing Regularity Clustering with Traditional Methods. In Boonthum-Denecke, Youngblood (Eds) Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2013, St. Pete Beach, Florida. May 22-24, 2013. AAAI Press 2013. pp 527-532
49. San Pedro, M., Baker, R., Bowers, A. & Heffernan, N. (2013). Predicting College Enrollment from Student Interaction with an Intelligent Tutoring System in Middle School. In S. D'Mello, R. Calvo, & A. Olney (Eds.) Proceedings of the 6th International Conference on Educational Data Mining (EDM2013). Memphis, TN. pp. 177-184.
50. San Pedro, M., Baker, R., Gowda, S., & Heffernan, N. (2013). Towards an Understanding of Affect and Knowledge from Student Interaction with an Intelligent Tutoring System. In Lane, Yacef, Motow & Pavlik (Eds) The Artificial Intelligence in Education Conference. Springer-Verlag. pp. 41-50.
51. San Pedro, M.O., Snow,E., Baker, R.S., McNamara, D., Heffernan, N. (2015). Exploring Dynamic Assessments of Affect, Behavior, and Cognition and Math State Test Achievement. To appear in *Proceedings of the 8th International Conference on Educational Data Mining.*
52. Trivedi, S., Pardos, Z. & Heffernan, N. (2011). Clustering Students to Generate an Ensemble to Improve Standard Test Score Predictions In Biswas et al. (Eds) *Proceedings of the Artificial Intelligence in Education Conference 2011.* pp. 328–336.
53. Trivedi, S., Pardos, Z., Sarkozy, G. & Heffernan, N. (2011). Spectral Clustering in Educational Data Mining. In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., and Stamper, J. (Eds.) *Proceedings of the 4th International Conference on Educational Data Mining.* pp. 129-138.
54. Trivedi, S. Pardos, Z., Sarkozy, G. & Heffernan, N. (2012). Co-Clustering by Bipartite Spectral Graph Partitioning for Out-Of-Tutor Prediction. 5th International Conference on Educational Data Mining. pp. 33-40.

55. Walonoski, J. & Heffernan, N.T. (2006). Detection and analysis of off-task gaming behavior in intelligent tutoring systems. In Ikeda, Ashley & Chan (Eds.). Proceedings of the Eighth International Conference on Intelligent Tutoring Systems. Springer-Verlag: Berlin. pp. 382-391. 2006.
56. Wang, Y. & Heffernan, N. (2011). The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. *The 24th International FLAIRS Conference* Nominated for Best Student Paper.
57. Wang, Y. & Heffernan, N. (2012). Leveraging First Response Time into the Knowledge Tracing Model. 5th International Conference on Educational Data Mining. pp. 176-179.
58. Wang, Y. & Heffernan, N. (2012). The Student Skill Model. 11th International Conference on Intelligent Tutoring Systems. Springer. pp 399-404
59. Wang, Y. & Heffernan, N. (2013). Extending Knowledge Tracing to allow Partial Credit: Using Continuous versus Binary Nodes. In Lane, Yacef, Motow & Pavlik (Eds) *The Artificial Intelligence in Education Conference.* Springer-Verlag. pp. 181-188.

Results of Prior NSF Funding for Heffernan's "Partnership Implementing Math and Science Education: Assisting Middle School Use of Tutoring Technology" grant (GK12-#0742503 $2 million).

1. Broderick, Z., O'Connor, C., Mulcahy, C., Heffernan, N. & Heffernan, C. (2011). Increasing Parent Engagement in Student Learning Using an Intelligent Tutoring System. *Journal of Interactive Learning Research, 22*(4), 523-550. Chesapeake, VA: AACE. Retrieved August 15, 2013, from http://www.editlib.org/p/34133.
2. Kehrer, P., Kelly, K. & Heffernan, N. (2013). Does Immediate Feedback While Doing Homework Improve Learning. In Boonthum-Denecke, Youngblood (Eds) Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2013, St. Pete Beach, Florida. May 22-24, 2013. AAAI Press 2013. p 542-545.
3. Kelly, K., Heffernan, N., D'Mello, S., Namias, J., & Strain, A. (2013). Adding Teacher-Created Motivational Video to an ITS. In Boonthum-Denecke, Youngblood (Eds) Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2013, St. Peters Beach, Florida. pp. 503-508.
4. Kelly, K., Heffernan, N., Heffernan, C., Goldman, S., Pellegrino, G. & Soffer, D. (2013). Estimating the Effect of Web-Based Homework. In Lane, Yacef, Motow & Pavlik (Eds) The Artificial Intelligence in Education Conference. Springer-Verlag. pp. 824-827.
5. Kim, R, Weitz, R., Heffernan, N. & Krach, N. (2009). Tutored Problem Solving vs. "Pure": Worked Examples In N. A. Taatgen & H. van Rijn (Eds.), Proceedings of the 31st Annual Conference of the Cognitive Science Society (pp. 3121-3126). Austin, TX: Cognitive Science Society.
6. Koedinger, K., McLaughlin, E. & Heffernan, N. (2010). A Quasi-Experimental Evaluation of an On-line Formative Assessment and Tutoring System. *Journal of Educational Computing Research. Baywood Publishing. 4,* 489 - 510.
7. Lang, C., Heffernan, N., Ostrow, K., & Wang, Y. (in press). The Impact of Incorporating Student Confidence Items into an Intelligent Tutor: A Randomized Controlled Trial. To be included in Proceedings of the 8th International Conference on Educational Data Mining. Madrid, Spain.
8. Mendicino, M., Razzaq, L. & Heffernan, N. T. (2009). Improving Learning from Homework Using Intelligent Tutoring Systems. *Journal of Research on Technology in Education (JRTE). 41*(3), 331-346.
9. Ostrow, K. & Heffernan, N. (in press). The Role of Student Choice Within Adaptive Tutoring. To be included in Proceedings of the 17th International Conference on Artificial Intelligence in Education. Madrid, Spain.

10. Ostrow, K. & Heffernan, N. T. (2014). Testing the Multimedia Principle in the Real World: A Comparison of Video vs. Text Feedback in Authentic Middle School Math Assignments. In Stamper, J., Pardos, Z., Mavrikis, M., McLaren, B.M. (eds.) Proceedings of the 7th International Conference on Educational Data Mining. London, United Kingdom, July 4-7. pp. 296-299.
11. Ostrow, K., Heffernan, N.T. & Heffernan, C. (in press). Blocking vs., Interleaving A Conceptual Replication Examining Single-Session Effects within Middle School Math Homework. *The 17th Proceedings of the Conference on Artificial Intelligence in Education,* Madrid, Spain.
12. Pardos, Z., Dailey, M. & Heffernan, N. (2011). Learning what works in ITS from non-traditional randomized controlled trial data. *The International Journal of Artificial Intelligence in Education. 21*, 47-63.
13. Razzaq, L. & Heffernan, N. (2009). To Tutor or Not to Tutor: That is the Question. In Dimitrova, Mizoguchi, du Boulay & Graesser (Eds.) *Proceedings of the 2009 Artificial Intelligence in Education Conference.* IOS Press. pp. 457-464.
14. Razzaq, L. & Heffernan, N. (2010). Hints: Is It Better to Give or Wait to be Asked? In Aleven, V., Kay, J & Mostow, J. (Eds) *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010) Part 1*. Springer. pp. 349-358.
15. Razzaq, L. & Heffernan, N.T. (2006). Scaffolding vs. hints in the Assistment system. In Ikeda, Ashley & Chan (Eds.). Proceedings of the Eighth International Conference on Intelligent Tutoring Systems. Springer-Verlag: Berlin. pp. 635-644.
16. Razzaq, L., Heffernan, N. T., Lindeman, R. W. (2007). What Level of Tutor Interaction is Best? In Luckin & Koedinger (Eds.) *Proceedings of the 13th Conference on Artificial Intelligence in Education.* pp 222-229.
17. Razzaq, L., Mendicino, M. & Heffernan, N. (2008). Comparing classroom problem-solving with no feedback to web-based homework assistance. In Woolf, Aimeur, Nkambou and Lajoie (Eds.) *Proceeding of the 9th International Conference on Intelligent Tutoring Systems.* pp. 426 -437.
18. Sao Pedro, M., Gobert, J., Heffernan, N. & Beck, J. (2009). In N.A. Taathen & H. van Rjin (Eds.) Comparing Pedagogical Approaches for Teaching the Control of Variables Strategy. *Proceedings of the 31st Annual Conference of the Cognitive Science Society* Austin, TX: Cognitive Science Society.
19. Shrestha, P., Wei, X., Maharjan, A., Razzaq, L., Heffernan, N.T., & Heffernan, C., (2009). Are Worked Examples an Effective Feedback Mechanism During Problem Solving? In N. A. Taatgen & H. van Rijn (Eds.), Proceedings of the 31st Annual Conference of the Cognitive Science Society (pp. 1294-1299). Austin, TX: Cognitive Science Society.
20. Singh, R., Saleem, M., Pradhan, P., Heffernan, C., Heffernan, N., *Razzaq*, L. *Dailey*, M. O'Connor, C. & Mulchay, C. (2011). Feedback during Web-Based Homework: The Role of Hints In Biswas et al. (Eds) *Proceedings of the Artificial Intelligence in Education Conference 2011.* pp. 328–336.
21. Soffer, D., Das, V., Pellegrino, G., Goldman, S., Heffernan, N., Heffernan, C.,& Dietz, K. (2014) Improving Long-term Retention of Mathematical Knowledge through Automatic Reassessment and Relearning. American Educational Research Association (AERA 2014) Conference. Division C - Learning and Instruction / Section 1c: Mathematics. Retrieved April 20, 2015, from https://goo.gl/TPy2RZ
22. Weitz, R., Salden, R, Kim, R. & Heffernan, N. T. (2010). Comparing Worked Examples and Tutored Problem Solving: Pure vs. Mixed Approaches. 32nd Annual Conference of the Cognitive Science Society. Pages 2877-2881 Retrieved Oct 10, 2014 from http://csjarchive.cogsci.rpi.edu/proceedings/2010/papers/0676/paper0676.pdf

Results of Prior NSF Funding for Whitehill's "INT: Collaborative Research: Detecting, Predicting and Remediating Student Affect and Grit Using Computer Vision" grant #1551594, (09/01/2016-08/31/2020, $749,983. (Principal investigator of the award: Dr. Ivon Arroyo.

1. Aung, A.M., & Whitehill, J. (2018a). Harnessing Label Uncertainty to Improve Modeling: An Application to Student Engagement Recognition. Under review.
2. Aung, A.M., & Whitehill, J. (2018b). Automatic Eye-Gaze Following for Classroom Observation Analysis. Work in progress. https://users.wpi.edu/~jrwhitehill/AungWhitehill_EyeGazeFollowig_TechReport_Jan2018.pdf
3. Jiang, H., Dykstra, K., & Whitehill, J. (2018). Predicting When Teachers Look at Their Students in 1-on-1 Tutoring Sessions. Under review.

Figure SD1: The homework assignment and existing item report used in class to review homework. The assignment has two problems; problem 1 is a straightforward systems of equations problem and problem 2 has five parts. Note the last problem, 2e, is an open response question (typical for EngageNY). The item report shows the results from 8 students. Take note of the common wrong answer of 0 for #1.

1. Solve for x
-3y=12x+9 ; -3y=x-9

2. A local ski Club needs to choose between two companies.
- **SnowBird Charter** Charges $300 plus $12 per person
- **Mountain Charter** Charges $15 per person

2a. If 72 members signed up for the trip, what would be the total transportation cost with **SnowBird Charter**?

2b. If 72 members signed up for the trip, what would be the total transportation cost with **Mountain Charter**?

2c. Write an equation that expresses c, the total cost using **SnowBird** Charter, in terms of p, the total number of club members who go on the trip.

2d Write an equation that expresses c, the total cost using **Mountain** Charter, in terms of p, the total number of club members who go on the trip.

2e. If the club members want to choose the less expensive of the two companies, which company should they choose? Justify your answer by explaining how the number of club members who go on the trip should affect their decision.

Student/Problem [Anonymize]	Average	#1	#2a	#2b	#2c	#2d	#2e
Problem Average Graph	27%	27%	87%	96%	87%	92%	N/A
Common Wrong Answers		0 (65%)					
Correct Answer(s)		*-18/11*	*1164*	*1080*	*12p+300*	*15p*	*N/A*
Gangi	0%	✗ 5	✗ 5	✗ 5	✗ 5	✗ 5	Mountain Charter sounds better than Snowbird
Lalit	60%	✗ 0	✗ 1000	✓ 1080	✓ 12p+300	✓ 15p	If you have more members then you are going to want to go with snowbird charter but if you have less members you are going to want mountain charter
Courtney	0%	✗ 0	✗ 1000	✗ 1800	✗ 312	✗ 15	They should choose Mountain Charter if 50 people go… Mountain Charter total cost is 750$ Snowbird Charter total cost is 900$ Mountain Charter is cheaper.
Carl	0%	✗ 0	✗ 1000	✗ 1800	✗ 1800	✗ p	I would choose the Mountain Charter because it is cheaper for both smaller and larger groups of people so it's a better deal.
Sachi	100%	✓ -18/11	✓ 1164	✓ 1080	✓ 12p+300	✓ 15p	It depends. The break even point for both companies is 100 members, where the cost is $1500 for bothe. If there were less than or equal to 100 members, the club should choose Mountain Charter. If there are more than or equal to 100 members, the club should choose Snowbird Charter.
Wei	80%	✗ 0	✓ 1164	✓ 1080	✓ 12p+300	✓ 15p	If I wanted to choose a less expensive of the two companies, I would choose MOUNTAIN CHARTER. If I had 4 people round trip i would have to pay $348 for SNOWBIRD CHARTER> But, if I choose to pay to MOUNTAIN CHARTER I would have to pay only $60 which $288 less than SNOWBIRD CHARTER. So i would choose MOUNTAIN CHARTER over SNOWBIRD CHARTER.
Wakeeta	100%	✓ -18/11	✓ 1164	✓ 1080	✓ 12p+300	✓ 15p	If 72 people go on the trip they should choose mountain charger because for 72 people it is cheaper than snowbird charter
Linda	80%	✓ -18/11	✗ 1000	✓ 1080	✓ 12p+300	✓ 15p	They should choose Mountain Charter. Since 72 members are going the cost would be 1080 as opposed to 1164 for snowbird. But if they had more members going the best choice would be Snowbird.

Figure SD2: The dialogue-initiation-interface. This is where the teacher goes to start a dialogue. For the 8 students the context-builder will select a student diagnosis (column 2) and then the dialogue-builder will select messages and actions for the teacher to select. The teacher can click on 'show' to see what content led to the diagnosis. The first message will be selected as a default. If the teacher decides not to send a dialogue start, then the teacher will be asked to "tell us why" (column 3) so the system can learn from the teachers decision.

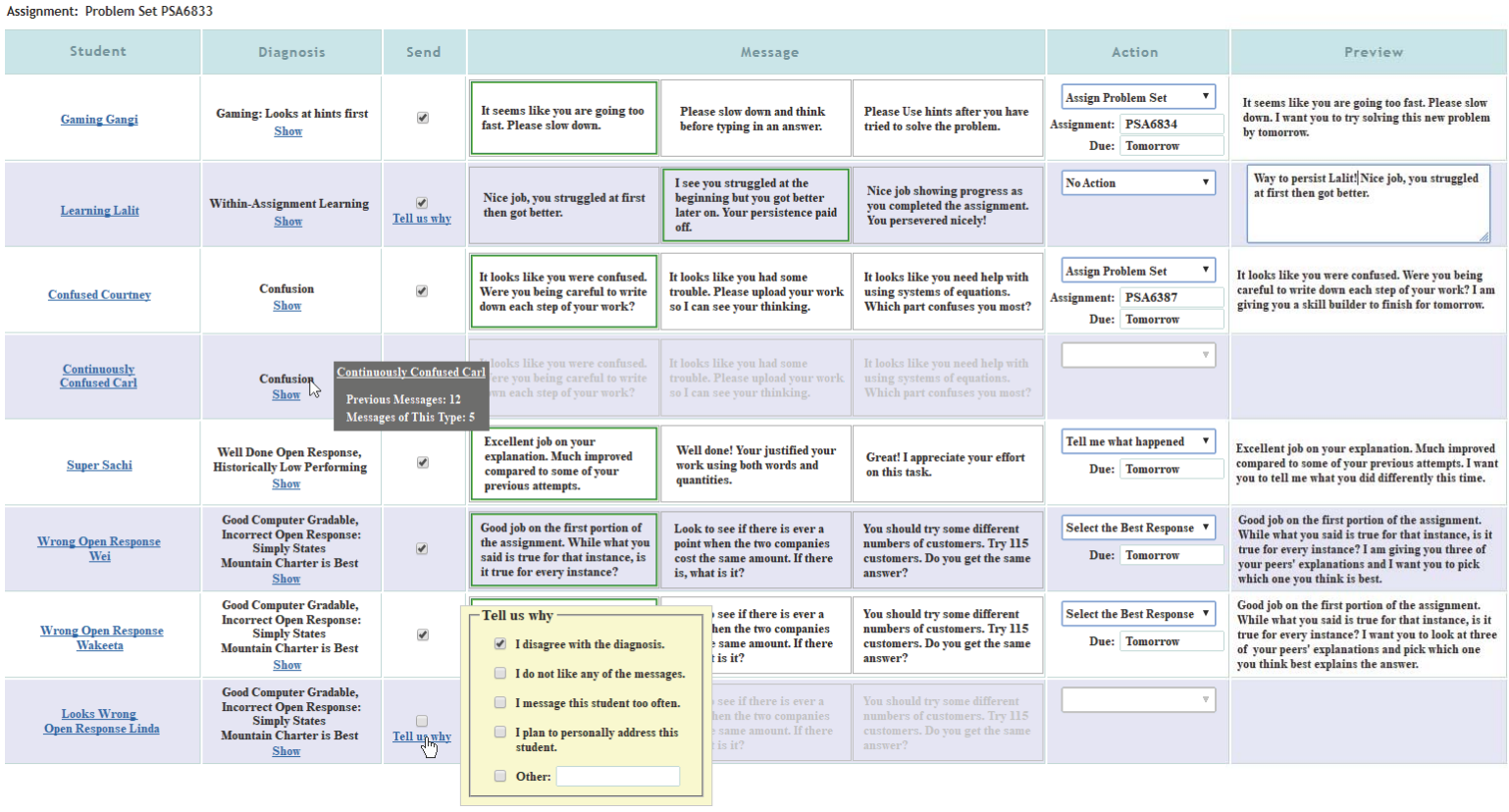

Figure SD3: After the teacher selects the message and action from the dialogue-initiation-interface the dialogue will begin. There will be a standard template for this dialogue start (center). It will include the student's name, a description of the context (in this example it is the series of actions the student performed as they did their homework that allowed DRIVER-SEAT to diagnose them as confused), the message, and the action required of the student. For Research Question 1, when we want to test to see if students respond better to dialogue starts with a personal teacher constructed format (left) or a generic computer constructed format (right).

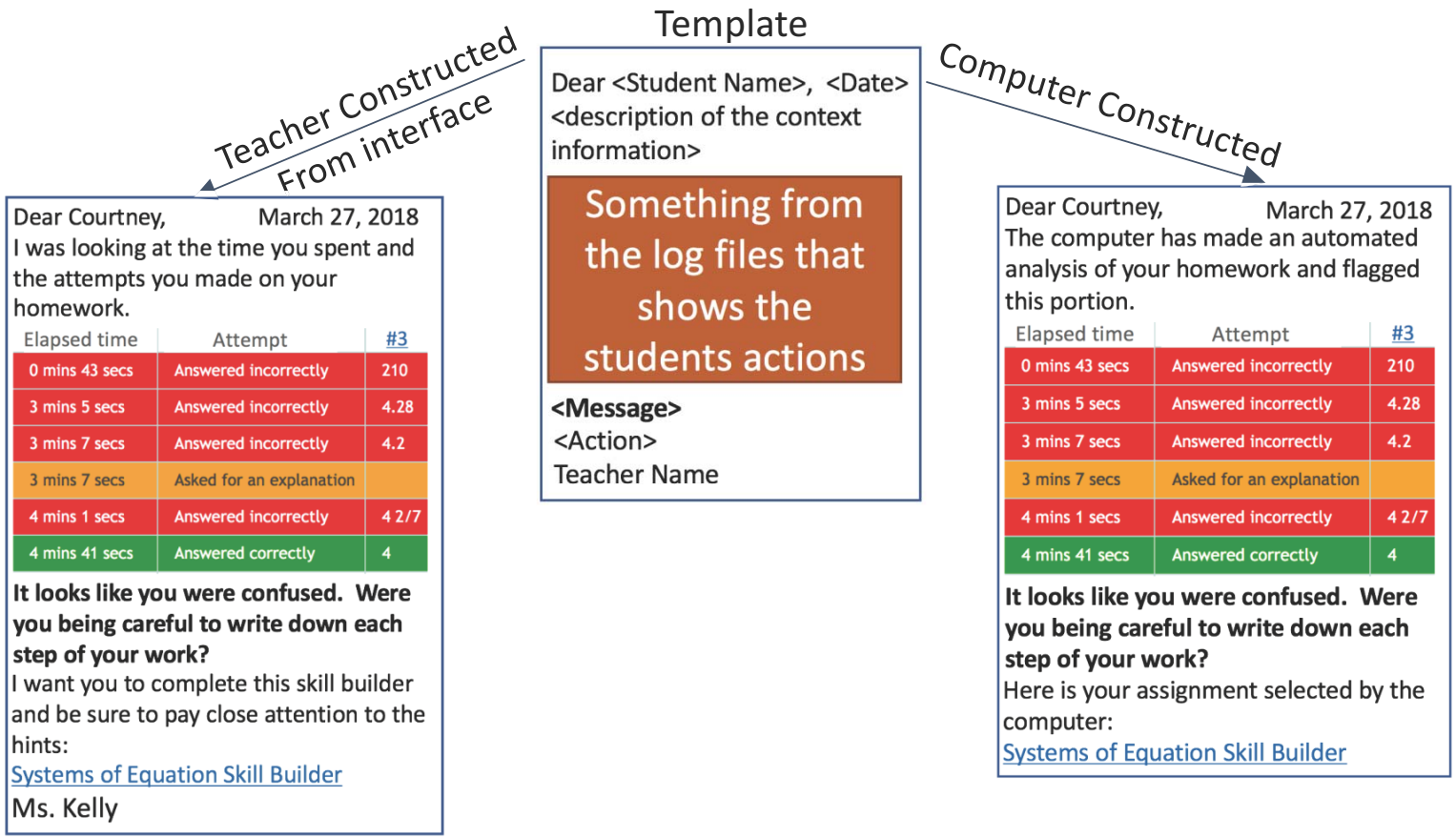

Figure SD4: In Stage 2 of the development process in year 1, the system will give suggestions of 3 diagnoses from which the teacher will choose one to focus. In this case, the teacher decided that Gaming was the most important diagnosis for Gangi. For Lalit, the teacher selected to focus on the evidence of learning and for Courtney the teacher opted for the default, confused.

Student	Diagnoses		
Gangi	Gaming Show	Confused Show	Poor open response Show
Lalit	Poor open response Show	Evidence of learning Show	Common wrong answer Show
Courtney	Confused Show	Gaming Show	Poor open response Show

Figure SD5: In Stage 3 of the development process in year 1, the system will automatically select a diagnosis accompanied by evidence, but then the system will ask the teacher to select an action. In this example the teacher kept the suggestions for Lalit and Courtney, but changed the suggestion for Gangi.

Student	Student Diagnosis (Detectors/Context Report)	Action		
Gangi	Gaming Show	Redo the assignment	Solve this new problem Select a problem	Inform parent
Lalit	Evidence of learning Show	None	Extra credit problem Select a problem	Inform parent
Courtney	Confused Show	Assign a Skill Builder Select a Skill Builder	Watch how-to video Select a video	Come get help after school

Chapter 8: HAND-RAISE and LIVE-CHART References

Alexander, J. M., Johnson, K. E., & Kelley, K. (2012). Longitudinal analysis of the relations between opportunities to learn about science and the development of interests related to science. *Science Education*, *96*(5), 763-786.

Barmby, P., Kind, P. M., & Jones, K. (2008). Examining changing attitudes in secondary school science. *International journal of science education*, 30(8), 1075-1093. Retrieved from https://www.tandfonline.com/doi/abs/10.1080/09500690701344966

Bernacki, M., Nokes-Malach, T., Richey, J. E., & Belenky, D. M. (2016). Science diaries: A brief writing intervention to improve motivation to learn science. *Educational Psychology*, *36*(1), 26-46. Retrieved from https://www.tandfonline.com/doi/abs/10.1080/01443410.2014.895293

Botelho, A. F., Baker, R. S., & Heffernan, N. T. (2017, June). Improving sensor-free affect detection using deep learning. In *International Conference on Artificial Intelligence in Education*(pp. 40-51). Springer, Cham. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-61425-0_4

Evolution of Student Interest in Science and Technology Studies: Policy Report. Paris. OECD. Retrieved from http://www.oecd.org/science/sci-tech/36645825.pd

George, R. (2006). A cross-domain analysis of change in students' attitudes toward science and attitudes about the utility of science. *International Journal of Science Education*, *28*(6), 571-589. Retrieved from https://eric.ed.gov/?id=EJ734637

Gottfried, A. E., Marcoulides, G. A., Gottfried, A. W., & Oliver, P. H. (2009). A latent curve model of parental motivational practices and developmental decline in math and science academic intrinsic motivation. *Journal of educational psychology*, *101*(3), 729.

Greenfield, T. A. (1997). Gender-and grade-level differences in science interest and participation. *Science education*, *81*(3), 259-276. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1098-237X(199706)81:3%3C259::AID-SCE1%3E3.0.CO;2-C

Guvercin, O., Tekkaya, C., & Sungur, S. (2010). A Cross Age Study of Elementary Students' Motivation towards Science Learning. *Hacettepe University Journal of Education*, *39*, 233-243. Retrieved from https://eric.ed.gov/?id=EJ916808

Holstein, K., Hong, G., Tegene, M., McLaren, B. M., & Aleven, V. (2018, March). 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. Retrieved from https://dl.acm.org/citation.cfm?id=3170377

Holstein, K., McLaren, B. M., & Aleven, V. (2017, March). 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. Retrieved from https://dl.acm.org/citation.cfm?id=3027451

Holstein, K., McLaren, B. M., & Aleven, V. (2018, June). Student learning benefits of a mixed-reality teacher awareness tool in AI-enhanced classrooms. In *International Conference on Artificial Intelligence in Education* (pp. 154-168). Springer, Cham. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-93843-1_12

Kelly, K., Heffernan, N., D'Mello, S., Namais, J., & Strain, A. (2013). Adding teacher-created motivational video to an ITS. In *Proceedings of 26th Florida Artificial Intelligence Research Society Conference* (pp. 503-508). Retrieved from http://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS13/paper/download/5934/6127

Krapp, A., & Prenzel, M. (2011). Research on interest in science: Theories, methods, and findings. *International journal of science education*, *33*(1), 27-50. Retrieved from https://www.tandfonline.com/doi/abs/10.1080/09500693.2010.518645

Mahoney, M. P. (2010). Students' Attitudes toward STEM: Development of an Instrument for High School STEM-Based Programs. *Journal of Technology Studies*, *36*(1), 24-34. Retrived from https://eric.ed.gov/?id=EJ906158

Makhlouf, J. & Mine, T. (2018) Predicting if students will pursue a STEM career using School-Aggregated Data from their usage of an Intelligent Tutoring System. *In Proceedings of the 11th International Conference on Educational Data Mining*. pp. 533-536. Retrieved from http://educationaldatamining.org/files/conferences/EDM2018/EDM2018_Preface_TOC_Proceedings.pdf

Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., & Heffernan, C. (2014). Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology*, *45*(3), 487-501.Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/bjet.12156

Osborne, J. & Dillon, J., 2008. *Science education in Europe: Critical Reflections*. Nuffield Foundation. Retrieved from http://www.nuffieldfoundation.org/sites/default/files/files/Sci_Ed_in_Europe_Report_Final(1).pdf

Ostrow, K. S. (2018). *A Foundation For Educational Research at Scale: Evolution and Application*. Retrieved from https://web.wpi.edu/Pubs/ETD/Available/etd-042418-205859/

Paquette, L., & Baker, R. S. (2017, June). Variations of gaming behaviors across populations of students and across learning environments. In *International Conference on Artificial Intelligence in Education* (pp. 274-286). Springer, Cham. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-61425-0_23

Potvin, P., & Hasni, A. (2014). Analysis of the decline in interest towards school science and technology from grades 5 through 11. *Journal of Science Education and Technology*, *23*(6), 784-802. Retrieved from https://link.springer.com/article/10.1007/s10956-014-9512-x

Roschelle, J., Feng, M., Murphy, R.F., & Mason, C.A. (2016). Online mathematics homework increases student achievement. *AERA Open, 2(4)*: 1-12. Retreived from http://journals.sagepub.com/doi/abs/10.1177/2332858416673968

San Pedro, M. O., Ocumpaugh, J., Baker, R. S., & Heffernan, N. T. (2014). Predicting STEM and Non-STEM College Major Enrollment from Middle School Interaction with Mathematics Educational Software. In Proceedings of the 7th International Conference on Educational Data Mining. pp. 276-279.

Selent, D. (June, 2017). *Creating Systems and Applying Large-Scale Methods to Improve Student Remediation in Online Tutoring Systems in Real-time and at Scale*. Retrieved from https://web.wpi.edu/Pubs/ETD/Available/etd-060817-000104/

Sjøberg, S., & Schreiner, C. (2005, December). How do learners in different cultures relate to science and technology? Results and perspectives from the project ROSE (the Relevance of Science Education). In *Asia-Pacific Forum on Science Learning and Teaching* (Vol. 6, No. 2, pp. 1-17). The Education University of Hong Kong, Department of Science and Environmental Studies. Retrieved from https://www.ied.edu.hk/apfslt/download/v6_issue2_files/foreword.pdf

Sorge, C. (2007). What Happens? Relationship of Age and Gender with Science Attitudes from Elementary to Middle School. *Science Educator*, *16*(2), 33-37.

References from Prior NSF Funding.

According to: http://www.nsf.gov/pubs/policydocs/pappguide/nsf14001/gpg_2.jsp, respondents must include "the publications resulting from the NSF award" and "a complete bibliographic citation for each publication must be provided either in this section or in the References Cited section of the proposal". Therefore, below are the references that resulted from prior NSF support (separated by each grant & CoPI for each CoPI that actually has Prior NSF Funding).

Results of Prior NSF Funding for Heffernan's "CAREER- Learning about Learning" grant (#0448319 $600,000).

1. Bahador, N., Pardos, Z., Heffernan & Baker, R. (2011). Less is More: Improving the Speed and Prediction Power of Knowledge Tracing by Using Less Data In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., and Stamper, J. (Eds.) *Proceedings of the 4th International Conference on Educational Data Mining.* pp. 101-110.
2. Baker, R., Pardos, Z., Gowda, S., Nooraei, B., & Heffernan, N. (2011). Ensembling Predictions of Student Knowledge within Intelligent Tutoring Systems. In Konstant et al. (Eds.) *20th International Conference on User Modeling, Adaptation and Personalization (UMAP 2011).* pp. 13-24.
3. Baker, R., Walonoski, J., Heffernan, T., Roll, I., Corbett, A. & Koedinger, K. (2008). Why students engage in "Gaming the System" behavior in interactive learning environments. *Journal of Interactive Learning Research (JILR).19*(2), 185-224.
4. Feng, M. & Heffernan, N. (2010). Can We Get Better Assessment From a Tutoring System Compared to Traditional Paper Testing? Can We Have Our Cake (Better Assessment) and Eat It Too (Student Learning During the Test) In Baker, R.S.J.d., Merceron, A., Pavlik, P.I. Jr. (Eds.) *Proceedings of the 3rd International Conference on Educational Data Mining*. pp. 41-50.
5. Feng, M., & Heffernan, N.T. (2006). Informing teachers live about student learning: Reporting in the Assistment system *Technology, Instruction, Cognition, and Learning Journal. 3*(1-2), 63.
6. Feng, M., & Heffernan, N.T. (2007). Towards live informing and automatic analyzing of student learning: Reporting in the Assistment system. *Journal of Interactive Learning Research (JILR) 18*(2), 207-230.
7. Feng, M., Beck, J., & Heffernan, N. (2009). Using Learning Decomposition and Bootstrapping with Randomization to Compare the Impact of Different Educational Interventions on Learning. In Barnes, Desmarais, Romero & Ventura (Eds) *Proc. of the 2nd International Conference on Educational Data Mining*. pp. 51-60.
8. Feng, M., Beck, J,. Heffernan, N. & Koedinger, K. (2008). Can an Intelligent Tutoring System Predict Math Proficiency as Well as a Standardized Test? In Baker & Beck (Eds.). *Proceedings of the 1st International Conference on Education Data Mining*. Montreal, Canada. pp. 107-116.
9. Feng, M., Heffernan, N. & Beck, J. (2009). Using Learning Decomposition to Analyze Instructional Effectiveness in the ASSISTment System. *Proceedings of the 2009 Artificial Intelligence in Education Conference*. IOS Press. pp. 523-530.
10. Feng, M., Heffernan, N., Beck, J, & Koedinger, K. (2008). Can we predict which groups of questions students will learn from? In Baker & Beck (Eds.). *Proceedings of the 1st International Conference on Education Data Mining*. Montreal, Canada. pp. 218-225.
11. Feng, M., Heffernan, N. & Koedinger, K.R. (2006a). Predicting state test scores better with intelligent tutoring systems: developing metrics to measure assistance required. In Ikeda, Ashley & Chan (Eds.).

Proceedings of the Eighth International Conference on Intelligent Tutoring Systems. Springer-Verlag: Berlin. pp. 31-40.

12. Feng, M., Heffernan, N. T., & Koedinger, K. R. (2006b). Addressing the testing challenge with a Web-based e-assessment system that tutors as it assesses. *Proceedings of the Fifteenth International World Wide Web Conference (WWW-06).* New York, NY: ACM Press. ISBN:1-59593-332-9. pp. 307-316.
13. Feng, M., Heffernan, N.T., & Koedinger, K.R. (2009). Addressing the assessment challenge in an Intelligent Tutoring System that tutors as it assesses. The Journal of *User Modeling and User-Adapted Interaction.19*, 243-266.
14. Feng, M., Heffernan, N. T., Mani, M., & Heffernan, C. (2007). Assessing students' performance longitudinally: Item difficulty parameter vs. skill learning tracking. The National Council on Educational Measurement 2007 Annual Conference, Chicago.
15. Feng, M., Heffernan, N.T., Heffernan, & C., Mani, M. (2009). Using Mixed-Effects Modeling to Analyze Different Grain-Sized Skill Models. IEEE Transactions on Learning Technologies, *2*(2), 79-92.
16. Gong, Y., Beck, J, Heffernan, N. (2010). Using Multiple Dirichlet distributions to improve parameter plausibility Educational Data Mining 2010. In Baker, R.S.J.d., Merceron, A., Pavlik, P.I. Jr. (Eds.) *Proceedings of the 3rd International Conference on Educational Data Mining.* pp. 61-70.
17. Gong, Y, Beck, J. E., Heffernan, N. T. (2011). How to Construct More Accurate Student Models: Comparing and Optimizing Knowledge Tracing and Performance Factor Analysis. *International Journal of Artificial Intelligence in Education*. *21*, 27-46.
18. Gong, Y., Beck, J. & Heffernan, N. (2010). Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting. In Aleven, V., Kay, J & Mostow, J. (Eds) *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010) Part 1*. Springer-Verlag, Berlin. pp. 35-44.
19. Gong, Y., Beck, J. & Heffernan, N. (2012). WEBsistments: Enabling an Intelligent Tutoring System to Excel at Explaining *Why* Other Than Showing *How*; 11th International Conference on Intelligent Tutoring Systems. Springer. pp 268-273
20. Gong, Y., Beck, J., Heffernan, N. & Forbes-Summers, E. (2010). The impact of gaming (?) on learning at the fine-grained level. In Aleven, V., Kay, J & Mostow, J. (Eds) *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010) Part 1*. Springer. pp.194-203.
21. Gong, Y., Rai, D. Beck, J. & Heffernan, N. (2009). Does Self-Discipline impact students' knowledge and learning? In Barnes, Desmarais, Romero & Ventura (Eds) *Proc. of the 2nd International Conference on Educational Data Mining.* pp. 61-70. ISBN: 978-84-613-2308-1.
22. Gowda, S., Baker, R.S.J.d., Pardos, Z., Heffernan, N. (2011). The Sum is Greater than the Parts: Ensembling Student Knowledge Models in ASSISTments. Proceedings of the KDD 2011 Workshop on KDD in Educational Data.
23. Hawkins, W., Baker, R. S. J. d., & Heffernan, N. T., (2013). Which is more responsible for boredom in intelligent tutoring systems: students (trait) or problems (state)? Affective Computing and Intelligent Interaction. Geneva. pp. 618-623.
24. Hawkins, W., Heffernan, N., Wang, Y. & Baker, S,J,d.. (2013). Extending the Assistance Model: Analyzing the Use of Assistance over Time. In S. D'Mello, R. Calvo, & A. Olney (Eds.) Proceedings of the 6th International Conference on Educational Data Mining (EDM2013). Memphis, TN. pp. 59-66.

25. Heffernan, N. & Heffernan, C. (2014). The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. *International Journal of Artificial Intelligence in Education. 24 (4),* 470-497.

26. Heffernan, N. T., Koedinger, K. & Razzaq, L. (2008). Expanding the model-tracing architecture: A 3rd generation intelligent tutor for Algebra symbolization. *The International Journal of Artificial Intelligence in Education. 18*(2), 153-178.

27. Heffernan N.T., Turner T. E., Lourenco A.L.N., Macasek M.A., Nuzzo-Jones G., & Koedinger K.R. (2006). The ASSISTment builder: Towards an analysis of cost effectiveness of ITS creation.*Proceedings of the 19th International FLAIRS Conference*, Melbourne Beach, Florida, USA. pp. 515-520.

28. Koedinger, K., McLaughlin, E. & Heffernan, N. (2010). A Quasi-Experimental Evaluation of an On-line Formative Assessment and Tutoring System. Journal of Educational Computing Research. Baywood Publishing. *4*, 489 - 510.

29. Militello, M., & Heffernan, N. (2009). Which one is "just right"? What educators should know about formative assessment systems. *International Journal of Educational Leadership Preparation, 4*(3), 1-8.

30. Ocumpaugh, J., Baker, R., Gowda, S., Heffernan, N., Heffernan, C. (2014). Population validity for Educational Data Mining models: A case study in affect detection. *British Journal of Educational Technology, 45* (3), 487-501.

31. Pardos, Z. & Heffernan, N. (2009). Detecting the Learning Value of Items in a Randomized Problem Set. In Dimitrova, Mizoguchi, du Boulay & Graesser (Eds.) *Proceedings of the 2009 Artificial Intelligence in Education Conference.* IOS Press. pp. 499-506.

32. Pardos, Z. & Heffernan, N. (2010). Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. In Paul De Bra, Alfred Kobsa, David Chin, (Eds.) *The 18th Proceedings of the International Conference on User Modeling, Adaptation and Personalization*. pp. 255-266.

33. Pardos, Z. & Heffernan, N. (2010). Navigating the parameter space of Bayesian Knowledge Tracing models: Visualization of the convergence of the Expectation Maximization algorithm. In Baker, R.S.J.d., Merceron, A., Pavlik, P.I. Jr. (Eds.) *Proceedings of the 3rd International Conference on Educational Data Mining*. pp. 161-170.

34. Pardos, Z. & Heffernan, N. (2011). KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In Konstant et al. (Eds.) *20th International Conference on User Modeling, Adaptation and Personalization (UMAP 2011).* pp. 243-254.

35. Pardos, Z. & Heffernan, N. (2012). Tutor Modeling vs. Student Modeling. Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference. Invited talk. Florida Artificial Intelligence Research Society (FLAIRS 2012). St. Peter Beach, Florida pp 420-425.

36. Pardos, Z. A., Beck, J., Ruiz, C. & Heffernan, N. T. (2008). The Composition Effect: Conjunctive or Compensatory? An Analysis of Multi-Skill Math Questions in ITS. In Baker & Beck (Eds.) *Proceedings of the First International Conference on Educational Data Mining.* Montreal, Canada. pp. 147-156.

37. Pardos, Z. A., Heffernan, N. T., Anderson, B. & Heffernan, C. (2007). The effect of model granularity on student performance prediction using Bayesian networks. The International User Modeling Conference 2007. pp. 435-439.

38. Pardos, Z., Gowda, S., Baker, R. & Heffernan, N. (2011). Ensembling Predictions of Student Post-Test Scores for an Intelligent Tutoring System. In Pechenizkiy, M., Calders, T., Conati, C., Ventura,

S., Romero , C., and Stamper, J. (Eds.) *Proceedings of the 4th International Conference on Educational Data Mining.* pp. 189-198.

39. Pardos, Z., Trivedi, S., Heffernan, N. & Sarkozy, G. (2012). Clustered Knowledge Tracing. 11th International Conference on Intelligent Tutoring Systems. pp 404-410
40. Pardos, Z.A., & Heffernan, N.T. (2009). Determining the Significance of Item Order In Randomized Problem Sets. In Barnes, Desmarais, Romero & Ventura (Eds.) *Proc. of the 2nd International Conference on Educational Data Mining*. pp. 111-120.
41. Pardos, Z.A., Gowda, S. M., Baker, R. S.J.D., Heffernan, N. T., (2012). The Sum is Greater than the Parts: Ensembling Models of Student Knowledge in Educational Software. *ACM's Knowledge Discovery and Datamining Explorations, 13*(2), 37-44
42. Qiu, Y., Pardos, *Z.* & Heffernan, N. (2012). Towards data driven user model improvement. Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference. Florida Artificial Intelligence Research Society (FLAIRS 2012). pp. 462-465.
43. Qiu, Y., Qi, Y., Lu, H., Pardos, Z. & Heffernan, N. (2011). Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., and Stamper, J. (Eds.) *Proceedings of the 4th International Conference on Educational Data Mining.* pp.139-148.
44. Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N.T., Koedinger, K. R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T.E., Upalekar. R, Walonoski, J.A., Macasek. M.A. & Rasmussen, K.P. (2005). The Assistment project: Blending assessment and assisting. In C.K. Looi, G. McCalla, B. Bredeweg, & J. Breuker (Eds.) *Proceedings of the 12th Artificial Intelligence in Education,* Amsterdam: ISO Press. pp. 555-562.
45. Razzaq, L., Heffernan, N., Feng, M., & Pardos Z. (2007). Developing Fine-Grained Transfer Models in the ASSISTment System. *Journal of Technology, Instruction, Cognition, and Learning. 5*(3), 289-304.
46. Razzaq, L., Mendicino, M. & Heffernan, N. (2008). Comparing classroom problem-solving with no feedback to web-based homework assistance. In Woolf, Aimeur, Nkambou and Lajoie (Eds.) *Proceeding of the 9th International Conference on Intelligent Tutoring Systems.* pp. 426 -437.
47. Razzaq, L., Patvarczki, J., Almeida, S.F., Vartak, M., Feng, M., Heffernan, N.T. and Koedinger, K. (2009). The ASSISTment builder: Supporting the Life-cycle of ITS Content Creation. IEEE Transactions on Learning Technologies Special Issue on Real-World Applications of Intelligent Tutoring Systems. *2*(2) 157-166.
48. Song, F., Trivedi, S., Wang, Y., Sárközy, G, & Heffernan, N. (2013). Applying Clustering to the Problem of Predicting Retention within an ITS: Comparing Regularity Clustering with Traditional Methods. In Boonthum-Denecke, Youngblood (Eds) Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2013, St. Pete Beach, Florida. May 22-24, 2013. AAAI Press 2013. pp 527-532
49. San Pedro, M., Baker, R., Bowers, A. & Heffernan, N. (2013). Predicting College Enrollment from Student Interaction with an Intelligent Tutoring System in Middle School. In S. D'Mello, R. Calvo, & A. Olney (Eds.) Proceedings of the 6th International Conference on Educational Data Mining (EDM2013). Memphis, TN. pp. 177-184.
50. San Pedro, M., Baker, R., Gowda, S., & Heffernan, N. (2013). Towards an Understanding of Affect and Knowledge from Student Interaction with an Intelligent Tutoring System. In Lane, Yacef, Motow & Pavlik (Eds) The Artificial Intelligence in Education Conference. Springer-Verlag. pp. 41-50.

51. San Pedro, M.O., Snow,E., Baker, R.S., McNamara, D., Heffernan, N. (2015). Exploring Dynamic Assessments of Affect, Behavior, and Cognition and Math State Test Achievement. To appear in *Proceedings of the 8th International Conference on Educational Data Mining.*

52. Trivedi, S., Pardos, Z. & Heffernan, N. (2011). Clustering Students to Generate an Ensemble to Improve Standard Test Score Predictions In Biswas et al. (Eds) *Proceedings of the Artificial Intelligence in Education Conference 2011.* pp. 328–336.

53. Trivedi, S., Pardos, Z., Sarkozy, G. & Heffernan, N. (2011). Spectral Clustering in Educational Data Mining. In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., and Stamper, J. (Eds.) *Proceedings of the 4th International Conference on Educational Data Mining.* pp. 129-138.

54. Trivedi, S. Pardos, Z., Sarkozy, G. & Heffernan, N. (2012). Co-Clustering by Bipartite Spectral Graph Partitioning for Out-Of-Tutor Prediction. 5th International Conference on Educational Data Mining. pp. 33-40.

55. Walonoski, J. & Heffernan, N.T. (2006). Detection and analysis of off-task gaming behavior in intelligent tutoring systems. In Ikeda, Ashley & Chan (Eds.). Proceedings of the Eighth International Conference on Intelligent Tutoring Systems. Springer-Verlag: Berlin. pp. 382-391. 2006.

56. Wang, Y. & Heffernan, N. (2011). The "Assistance" Model: Leveraging How Many Hints and Attempts a Student Needs. *The 24th International FLAIRS Conference* Nominated for Best Student Paper.

57. Wang, Y. & Heffernan, N. (2012). Leveraging First Response Time into the Knowledge Tracing Model. 5th International Conference on Educational Data Mining. pp. 176-179.

58. Wang, Y. & Heffernan, N. (2012). The Student Skill Model. 11th International Conference on Intelligent Tutoring Systems. Springer. pp 399-404

59. Wang, Y. & Heffernan, N. (2013). Extending Knowledge Tracing to allow Partial Credit: Using Continuous versus Binary Nodes. In Lane, Yacef, Motow & Pavlik (Eds) *The Artificial Intelligence in Education Conference.* Springer-Verlag. pp. 181-188.

Results of Prior NSF Funding for Heffernan's "Partnership Implementing Math and Science Education: Assisting Middle School Use of Tutoring Technology" grant (GK12-#0742503 $2 million).

1. Broderick, Z., O'Connor, C., Mulcahy, C., Heffernan, N. & Heffernan, C. (2011). Increasing Parent Engagement in Student Learning Using an Intelligent Tutoring System. *Journal of Interactive Learning Research, 22*(4), 523-550. Chesapeake, VA: AACE. Retrieved August 15, 2013, from http://www.editlib.org/p/34133.

2. Kehrer, P., Kelly, K. & Heffernan, N. (2013). Does Immediate Feedback While Doing Homework Improve Learning. In Boonthum-Denecke, Youngblood (Eds) Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2013, St. Pete Beach, Florida. May 22-24, 2013. AAAI Press 2013. p 542-545.

3. Kelly, K., Heffernan, N., D'Mello, S., Namias, J., & Strain, A. (2013). Adding Teacher-Created Motivational Video to an ITS. In Boonthum-Denecke, Youngblood (Eds) Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2013, St. Peters Beach, Florida. pp. 503-508.

4. Kelly, K., Heffernan, N., Heffernan, C., Goldman, S., Pellegrino, G. & Soffer, D. (2013). Estimating the Effect of Web-Based Homework. In Lane, Yacef, Motow & Pavlik (Eds) The Artificial Intelligence in Education Conference. Springer-Verlag. pp. 824-827.

5. Kim, R, Weitz, R., Heffernan, N. & Krach, N. (2009). Tutored Problem Solving vs. "Pure": Worked Examples In N. A. Taatgen & H. van Rijn (Eds.), Proceedings of the 31st Annual Conference of the Cognitive Science Society (pp. 3121-3126). Austin, TX: Cognitive Science Society.

6. Koedinger, K., McLaughlin, E. & Heffernan, N. (2010). A Quasi-Experimental Evaluation of an On-line Formative Assessment and Tutoring System. *Journal of Educational Computing Research. Baywood Publishing. 4,* 489 - 510.

7. Lang, C., Heffernan, N., Ostrow, K., & Wang, Y. (in press). The Impact of Incorporating Student Confidence Items into an Intelligent Tutor: A Randomized Controlled Trial. To be included in Proceedings of the 8th International Conference on Educational Data Mining. Madrid, Spain.

8. Mendicino, M., Razzaq, L. & Heffernan, N. T. (2009). Improving Learning from Homework Using Intelligent Tutoring Systems. *Journal of Research on Technology in Education (JRTE). 41*(3), 331-346.

9. Ostrow, K. & Heffernan, N. (in press). The Role of Student Choice Within Adaptive Tutoring. To be included in Proceedings of the 17th International Conference on Artificial Intelligence in Education. Madrid, Spain.

10. Ostrow, K. & Heffernan, N. T. (2014). Testing the Multimedia Principle in the Real World: A Comparison of Video vs. Text Feedback in Authentic Middle School Math Assignments. In Stamper, J., Pardos, Z., Mavrikis, M., McLaren, B.M. (eds.) Proceedings of the 7th International Conference on Educational Data Mining. London, United Kingdom, July 4-7. pp. 296-299.

11. Ostrow, K., Heffernan, N.T. & Heffernan, C. (in press). Blocking vs., Interleaving A Conceptual Replication Examining Single-Session Effects within Middle School Math Homework. *The 17th Proceedings of the Conference on Artificial Intelligence in Education,* Madrid, Spain.

12. Pardos, Z., Dailey, M. & Heffernan, N. (2011). Learning what works in ITS from non-traditional randomized controlled trial data. *The International Journal of Artificial Intelligence in Education. 21*, 47-63.

13. Razzaq, L. & Heffernan, N. (2009). To Tutor or Not to Tutor: That is the Question. In Dimitrova, Mizoguchi, du Boulay & Graesser (Eds.) *Proceedings of the 2009 Artificial Intelligence in Education Conference.* IOS Press. pp. 457-464.

14. Razzaq, L. & Heffernan, N. (2010). Hints: Is It Better to Give or Wait to be Asked? In Aleven, V., Kay, J & Mostow, J. (Eds) *Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS2010) Part 1*. Springer. pp. 349-358.

15. Razzaq, L. & Heffernan, N.T. (2006). Scaffolding vs. hints in the Assistment system. In Ikeda, Ashley & Chan (Eds.). Proceedings of the Eighth International Conference on Intelligent Tutoring Systems. Springer-Verlag: Berlin. pp. 635-644.

16. Razzaq, L., Heffernan, N. T., Lindeman, R. W. (2007). What Level of Tutor Interaction is Best? In Luckin & Koedinger (Eds.) *Proceedings of the 13th Conference on Artificial Intelligence in Education.* pp 222-229.

17. Razzaq, L., Mendicino, M. & Heffernan, N. (2008). Comparing classroom problem-solving with no feedback to web-based homework assistance. In Woolf, Aimeur, Nkambou and Lajoie (Eds.) *Proceeding of the 9th International Conference on Intelligent Tutoring Systems.* pp. 426 -437.

18. Sao Pedro, M., Gobert, J., Heffernan, N. & Beck, J. (2009). In N.A. Taathen & H. van Rjin (Eds.) Comparing Pedagogical Approaches for Teaching the Control of Variables Strategy. *Proceedings of the 31st Annual Conference of the Cognitive Science Society* Austin, TX: Cognitive Science Society.

19. Shrestha, P., Wei, X., Maharjan, A., Razzaq, L., Heffernan, N.T., & Heffernan, C., (2009). Are Worked Examples an Effective Feedback Mechanism During Problem Solving? In N. A. Taatgen & H. van Rijn (Eds.), Proceedings of the 31st Annual Conference of the Cognitive Science Society (pp. 1294-1299). Austin, TX: Cognitive Science Society.

20. Singh, R., Saleem, M., Pradhan, P., Heffernan, C., Heffernan, N., *Razzaq*, L. *Dailey*, M. O'Connor, C. & Mulchay, C. (2011). Feedback during Web-Based Homework: The Role of Hints In Biswas et al. (Eds) *Proceedings of the Artificial Intelligence in Education Conference 2011.* pp. 328–336.

21. Soffer, D., Das, V., Pellegrino, G., Goldman, S., Heffernan, N., Heffernan, C.,& Dietz, K. (2014) Improving Long-term Retention of Mathematical Knowledge through Automatic Reassessment and Relearning. American Educational Research Association (AERA 2014) Conference. Division C - Learning and Instruction / Section 1c: Mathematics. Retrieved April 20, 2015, from https://goo.gl/TPy2RZ

22. Weitz, R., Salden, R, Kim, R. & Heffernan, N. T. (2010). Comparing Worked Examples and Tutored Problem Solving: Pure vs. Mixed Approaches. 32nd Annual Conference of the Cognitive Science Society. Pages 2877-2881 Retrieved Oct 10, 2014 from http://csjarchive.cogsci.rpi.edu/proceedings/2010/papers/0676/paper0676.pdf