This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2912162, IEEE Transactions on Learning Technologies

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

Anthony F. Botelho, Ashvini Varatharaj, Thanaporn Patikorn, Diana Doherty,
Seth A. Adjei, and Joseph E. Beck

***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.**

***Index Terms*—Early detection, Wheel Spinning, Stopout, Transfer learning, Deep learning , Persistence**

I. 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 [1]. 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 [2], perseverance [3], and productive failure [4]. 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" [5]. 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2912162, IEEE Transactions on Learning Technologies

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.

II. BACKGROUND

A. Wheel Spinning

Several previous works have explored and have attempted to model student wheel spinning behavior in several platforms including Cognitive tutor [6] and ASSISTments [5] [7], while other work has explored policies to help prevent wheel spinning [8]. 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 [5] 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 [7]. 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 hypothesize 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.

B. 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) [9] [10] [11] [12] [13]. 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 [14]. 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 [15] and even transfer across courses [16]. 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 [17]), 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 [18], 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.

C. 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) [19], 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 [20] [21] [22], predicting student graduation [15] and real-time performance [16] in MOOCs, detecting student affective state [23], and predicting long-term outcomes [24] [25].

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 [26]). 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. [27]) or implicitly (e.g. [25]), that better describe the data to make better-informed model estimates. In this way, such models also support the application of transfer learning [28] to better understand the relationship between outcomes of interest by providing the means to observe how learned features generalize across prediction tasks.

III. 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 [29]. 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).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2912162, IEEE Transactions on Learning Technologies

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 I: Description of the generated action-level features.

A. 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 I.

B. 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 assignment 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 [5] [7]. 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.

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 II: The notable descriptives of the dataset.

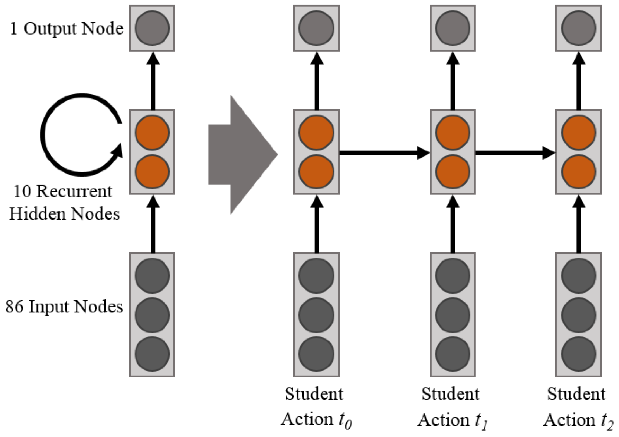


Fig. 1: A simplified representation of the LSTM model structure, illustrating how information flows from previous timesteps to inform each model estimate.

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, 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 II, contains over 100 thousand student assignments from over 12 thousands students, resulting in approximately 1 million actions to be used by our models.

IV. 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 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.

A. 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 [30], 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 [5] [7]. 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 [31] 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 [32] in Python with a 3 layer structure; the input layer included 86 nodes corresponding with each of the available

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2912162, IEEE Transactions on Learning Technologies

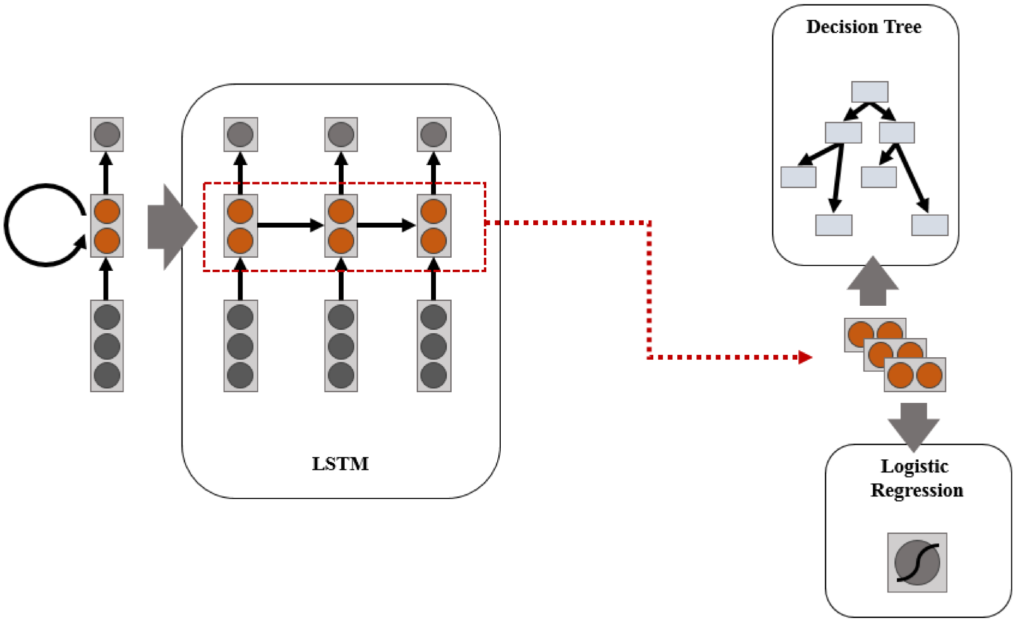


Fig. 2: A visual example of the transfer learning procedure. The hidden layer of the trained LSTM model is used as input to train each a decision tree and logistic regression to predict each wheel spinning and stopout behavior.

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 [33], 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.

B. 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 [34] [35]. 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 I) 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 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.

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2912162, IEEE Transactions on Learning Technologies

	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 III: 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 IV: Predicting Stopout in current assignment

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.

V. Results

A. 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 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.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2912162, IEEE Transactions on Learning Technologies

	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 V: 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 VI: Predicting Stopout in next assignment

B. Model Performance

Our results are recorded such that each of the Tables III-VI record results of one outcome variable. Table III 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 I). 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 III 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 III, Table IV records the model performance for predicting if a student is going to stopout in the current assignment. The order is similar to Table III 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 V 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2912162, IEEE Transactions on Learning Technologies

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 VI 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 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 III, 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.

C. 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 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 [36] for the computed recall value at each opportunity. The confidence bounds for AUC is computed using pROC [37], 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 and 6 respectively; we compare these, then to the model performance for each within-assignment wheel spinning and stopout depicted in Figures 3 and 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 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, 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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2912162, IEEE Transactions on Learning Technologies

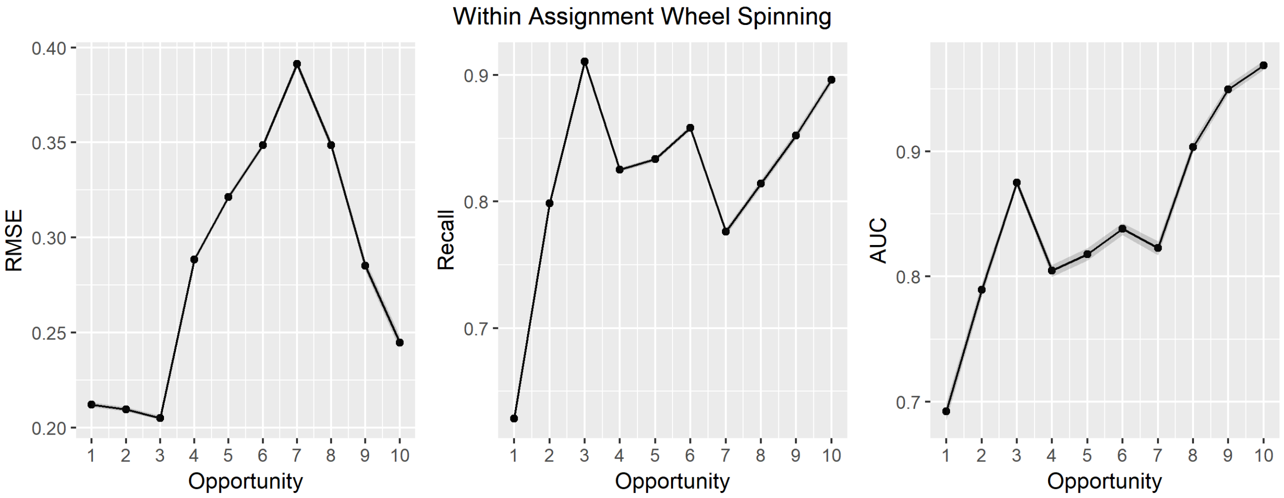


Fig. 3: The performance of the LSTM model in predicting within-assignment wheel spinning by opportunity.

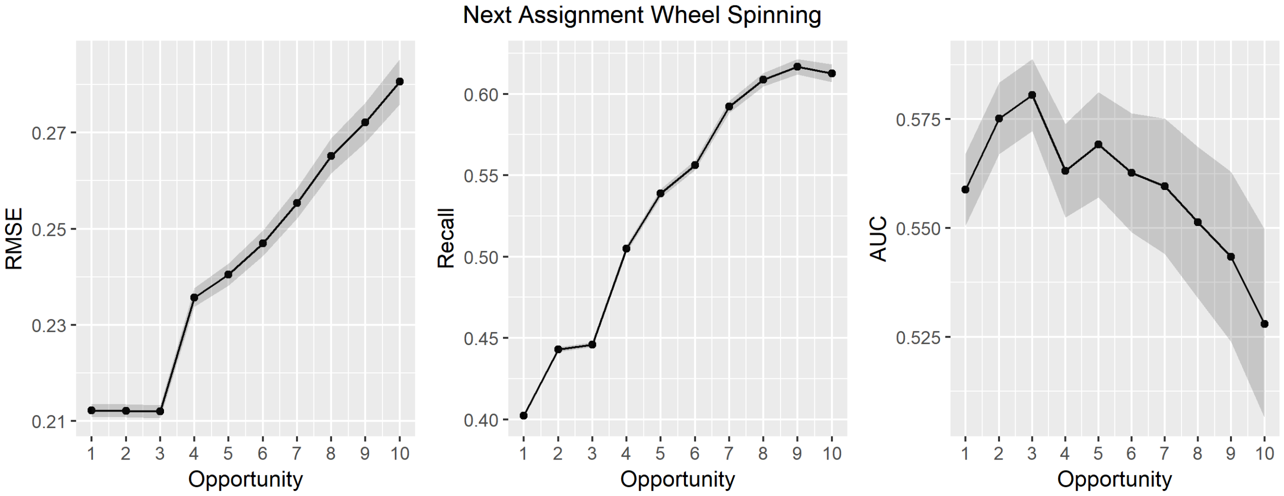


Fig. 4: The performance of the LSTM model in predicting next assignment wheel spinning by opportunity.

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 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 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.

The performance of the LSTM model in predicting next assignment stopout, as depicted in Figure 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,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TLT.2019.2912162, IEEE Transactions on Learning Technologies

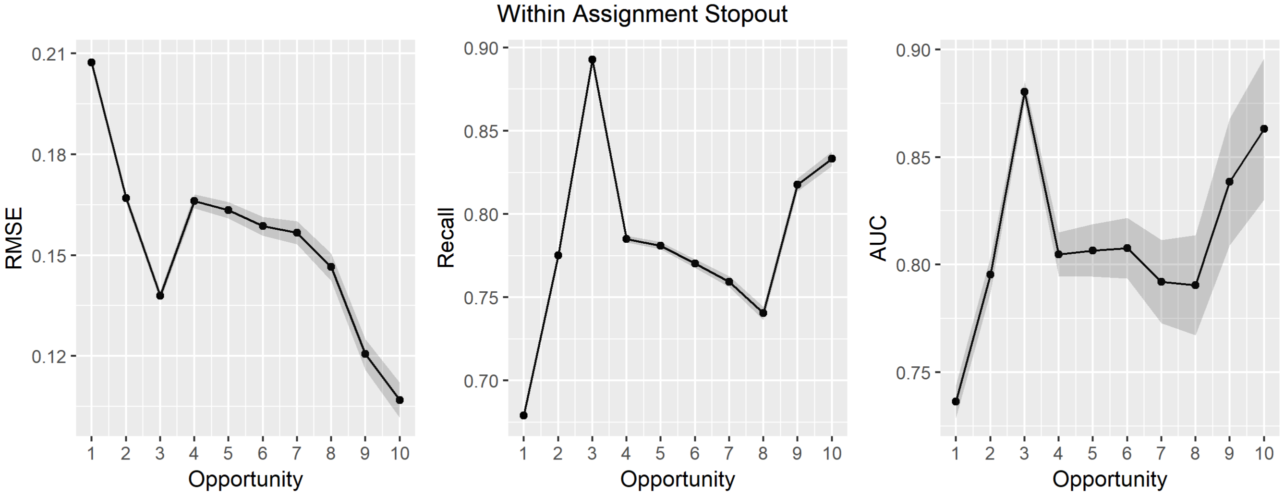


Fig. 5: The performance of the LSTM model in predicting within-assignment stopout by opportunity.

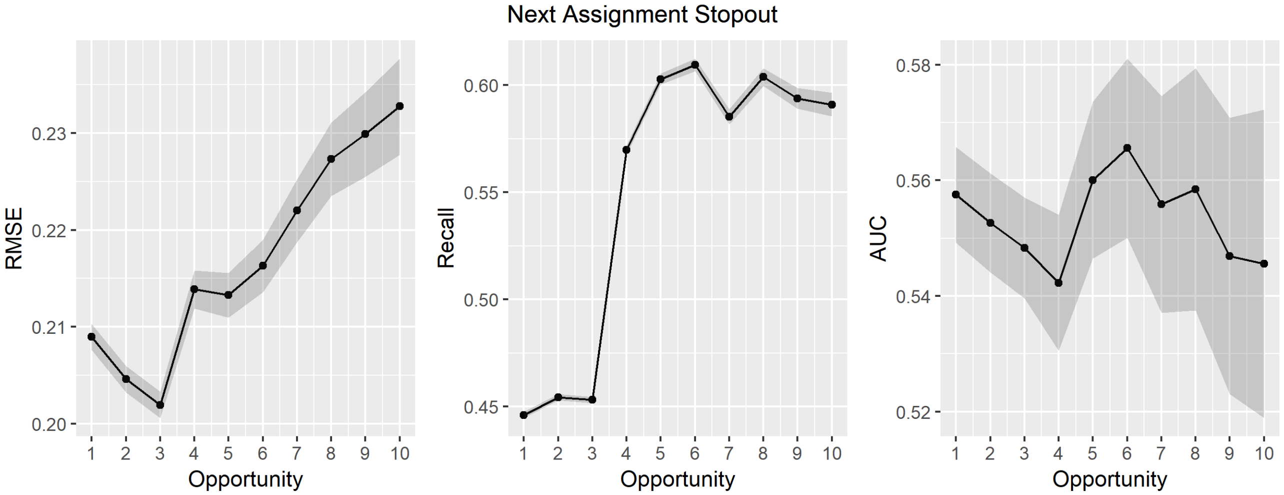


Fig. 6: The performance of the LSTM model in predicting next assignment stopout by opportunity.

it would appear that, similar to the AUC of the next assignment wheel spinning model illustrated in Figure 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 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.

VI. 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 [38])? 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?

VII. 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.

Acknowledgment

We thank multiple current NSF grants (ACI-1440753, DRL-1252297, DRL-1109483, DRL-1316736, DGE-1535428, OAC-1724889,OAC-1636782 & DRL-1031398) , the US Dept. of Ed (IES R305A180401, R305A120125 & R305C100024 and GAANN), and the ONR.

References

[1] R. D. Roscoe and M. T. Chi, "Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors explanations and questions," *Review of Educational Research*, vol. 77, no. 4, pp. 534–574, 2007.

[2] A. L. Duckworth, C. Peterson, M. D. Matthews, and D. R. Kelly, "Grit: perseverance and passion for long-term goals." *Journal of personality and social psychology*, vol. 92, no. 6, p. 1087, 2007.

[3] C. Peterson, M. E. Seligman *et al.*, *Character strengths and virtues: A handbook and classification*. Oxford University Press, 2004, vol. 1.

[4] M. Kapur, "Productive failure," *Cognition and instruction*, vol. 26, no. 3, pp. 379–424, 2008.

[5] J. E. Beck and Y. Gong, "Wheel-spinning: Students who fail to master a skill," in *International Conference on Artificial Intelligence in Education*. Springer, 2013, pp. 431–440.

[6] N. Matsuda, S. Chandrasekaran, and J. C. Stamper, "How quickly can wheel spinning be detected?" in *EDM*, 2016, pp. 607–608.

[7] Y. Gong and J. E. Beck, "Towards detecting wheel-spinning: Future failure in mastery learning," in *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*. ACM, 2015, pp. 67–74.

[8] T. Käser, S. Klingler, and M. Gross, "When to stop?: towards universal instructional policies," in *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*. ACM, 2016, pp. 289–298.

[9] D. S. Chaplot, E. Rhim, and J. Kim, "Predicting student attrition in moocs using sentiment analysis and neural networks." in *AIED Workshops*, 2015.

[10] W. Xing, X. Chen, J. Stein, and M. Marcinkowski, "Temporal predication of dropouts in moocs: Reaching the low hanging fruit through stacking generalization," *Computers in Human Behavior*, vol. 58, pp. 119–129, 2016.

[11] D. Yang, T. Sinha, D. Adamson, and C. P. 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*, vol. 11, 2013, p. 14.

[12] C. P. Rosé, R. Carlson, D. Yang, M. Wen, L. Resnick, P. Goldman, and J. Sherer, "Social factors that contribute to attrition in moocs," in *Proceedings of the first ACM conference on Learning@ scale conference*. ACM, 2014, pp. 197–198.

[13] A. Lamb, J. Smilack, A. Ho, and J. 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*. ACM, 2015, pp. 21–30.

[14] R. F. Kizilcec and S. Halawa, "Attrition and achievement gaps in online learning," in *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*. ACM, 2015, pp. 57–66.

[15] B.-H. Kim, E. Vizitei, and V. Ganapathi, "Gritnet: Student performance prediction with deep learning," *arXiv preprint arXiv:1804.07405*, 2018.

[16] ——, "Gritnet 2: Real-time student performance prediction with domain adaptation," *arXiv preprint arXiv:1809.06686*, 2018.

[17] A. Botelho, H. Wan, and N. Heffernan, "The prediction of student first response using prerequisite skills," in *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*. ACM, 2015, pp. 39–45.

[18] J. Heckman, N. Hohmann, J. Smith, and M. Khoo, "Substitution and dropout bias in social experiments: A study of an influential social experiment," *The Quarterly Journal of Economics*, vol. 115, no. 2, pp. 651–694, 2000.

[19] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[20] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," in *Advances in Neural Information Processing Systems*, 2015, pp. 505–513.

[21] M. Khajah, R. V. Lindsey, and M. C. Mozer, "How deep is knowledge tracing?" in *Proceedings of the 9th International Conference on Educational Data Mining*, 2016, pp. 94–101.

[22] X. Xiong, S. Zhao, E. Van Inwegen, and J. Beck, "Going deeper with deep knowledge tracing." in *Proceedings of the 9th International Conference on Educational Data Mining*, 2016, pp. 545–550.

[23] A. F. Botelho, R. S. Baker, and N. T. Heffernan, "Improving sensor-free affect detection using deep learning," in *International Conference on Artificial Intelligence in Education*. Springer, 2017, pp. 40–51.

[24] A. C. Sales, A. Botelho, T. Patikorn, and N. 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*, 2018, pp. 479–486,.

[25] C.-k. Yeung, Z. Lin, K. Yang, and D.-y. Yeung, "Incorporating features learned by an enhanced deep knowledge tracing model for stem/non-stem job prediction," *arXiv preprint arXiv:1806.03256*, 2018.

[26] A. F. Botelho, R. S. Baker, J. Ocumpaugh, and N. T. Heffernan, "Studying affect dynamics and chronometry using sensor-free detectors." in *Proceedings of the 11th International Conference on Educational Data Mining*, 2018, pp. 157–166.

[27] L. Zhang, X. Xiong, S. Zhao, A. Botelho, and N. T. Heffernan, "Incorporating rich features into deep knowledge tracing," in *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*. ACM, 2017, pp. 169–172.

[28] L. Y. Pratt, "Discriminability-based transfer between neural networks," in *Advances in neural information processing systems*, 1993, pp. 204–211.

[29] N. T. Heffernan and C. L. 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*, vol. 24, no. 4, pp. 470–497, 2014.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] 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*, vol. 12, pp. 2825–2830, 2011.

[32] M. Abadi, A. Agarwal, P. Barham, ..., and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[35] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, vol. 1341, no. 3, p. 1, 2009.

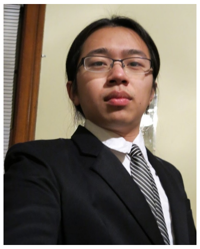
[36] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, 1927.

[37] X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez, and M. Müller, "proc: an open-source package for r and s+ to analyze and compare roc curves," *BMC bioinformatics*, vol. 12, no. 1, p. 77, 2011.

[38] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

Anthony F. Botelho Anthony is a Learning Sciences and Technologies PhD student at Worcester Polytechnic Institute and works as a member of the ASSISTments team to help develop and perform research within the online learning platform.

Ashvini Varatharaj is a Masters student in the Computer Science Department in Worcester Polytechnic Institute. She is a Research Assistant in the ASSISTments lab and a student of Neil Heffernan.



Thanaporn Patikorn is a PhD student at Worcester Polytechnic Institute majoring in Computer Science. He is a Research Assistant in the ASSISTments lab and a student of Neil Heffernan.

Diana Doherty completed her Undergraduate degree in Computer Science at Worcester Polytechnic Institute in the Fall of 2018.

Seth A. Adjei is a visiting assistant professor at Northern Kentucky University. His research has been focused on finding ways in which to improve student learning and to help low-achieving students benefit from interventions that increase their achievement, particularly in mathematics.

Joseph E. Beck is an assistant professor of Computer Science at WPI. His research focuses on educational data mining, a new discipline that develops techniques for analyzing large educational data sets to make discoveries that will improve teaching and learning. His work centers on estimating how computer tutors impact learning.