Early detection of wheel spinning: Comparison across tutors, models, features, and operationalizations

Chuankai Zhang
Carnegie Mellon University

Yanzun Huang
Carnegie Mellon University

Jingyu Wang
Carnegie Mellon University

Dongyang Lu
Carnegie Mellon University

Weiqi Fang
Carnegie Mellon University

John Stamper
Carnegie Mellon University

Stephen Fancsali
Carnegie Mellon University

Kenneth Holstein
Carnegie Mellon University

Vincent Aleven
Carnegie Mellon University

In Proceedings of the 12th International
Conference on Educational Data Mining
Montreal, QC, Canada
July 2 - 5, 2019

Early Detection of Wheel Spinning: Comparison across Tutors, Models, Features, and Operationalizations

Chuankai Zhang, Yanzun Huang, Jingyu Wang, Dongyang Lu, Weiqi Fang, John Stamper, Stephen Fancsali, Kenneth Holstein, Vincent Aleven
Carnegie Mellon University, Carnegie Learning, Inc.
{chuankaz, yanzunh, jingyuw1, dongyanl, weiqif}@andrew.cmu.edu,
{jstamper, kjholste, aleven}@cs.cmu.edu, sfancsali@carnegielearning.com

ABSTRACT

"Wheel spinning" is the phenomenon in which a student fails to master a Knowledge Component (KC), despite significant practice. Ideally, an intelligent tutoring system would detect this phenomenon early, so that the system or a teacher could try alternative instructional strategies. Prior work has put forward several criteria for wheel spinning and has demonstrated that wheel spinning can be detected reasonably early. Yet the literature lacks systematic comparisons among the multiple wheel spinning criteria, features, and models that have been proposed, across multiple evaluation criteria (e.g., earliness, precision, and generalizability) and datasets. In our experiments, we constructed six wheel spinning detectors and compared their performance under two different wheel spinning criteria with three datasets. The results show that two prominent criteria for wheel spinning diverge substantially, and that a Random Forest model has the most consistent performance in early detection of wheel spinning across datasets and wheel spinning criteria. In addition, we found that a simple model overlooked by previous research (Logistic Regression trained on a single feature) is able to detect wheel spinning at an early stage with decent performance. This work brings us closer to unifying strands of prior work on wheel spinning (e.g., understanding how different criteria compare) and to early detection of wheel spinning in educational practice.

Keywords

wheel spinning, student modeling, intelligent tutors

1. INTRODUCTION

Intelligent tutoring systems (ITS) aim to guide students towards mastery of knowledge components by providing step-by-step personalized guidance. However, there are cases where students persistently work on problems without making progress towards mastery. This phenomenon of unproductive student persistence has been called "wheel spinning" [1]. If ITSs were able to detect potential wheel spinning as early as possible, they might be able to adjust their instructional strategies accordingly to avoid wasting students' time.

Beck & Gong [1] operationalized wheel spinning as failing to get three attempts correct in a row within the first 10 practice opportunities. We refer to this as "three correct in a row criterion." They presented evidence in [3] that wheel spinning is not a rare phenomenon. Other operationalizations of unproductive persistence have since been proposed. For example, Predictive Stability (PS) is a when-to-stop policy for ITSs proposed in [5], which stops when the probability of a student getting the next step correct stabilizes. The policy uses student performance at each step to decide whether the ITS should stop giving more questions, either because of mastery or wheel spinning. The Predictive Stability++ (PS++) policy [5] provides further analysis about mastery *after* the PS policy would have stopped. This policy can detect wheel spinning under various student models. Although the two operationalizations are, at the surface, rather different, prior work has not investigated how they compare.

More generally, although prior work has introduced various machine learning models for the early detection of wheel spinning, fitted on a variety of datasets, we are not aware of any systematic comparison across models, datasets, and operationalizations. This makes it difficult for researchers to compare and establish global evaluations, which is important both for practical and theoretical reasons. Therefore, in this study, we conducted a comprehensive examination to address the following questions: (1) To what extent do different operationalizations of wheel spinning agree or disagree? (2) Which set of features leads to better predictions? (3) What is the simplest set of features that can be effective? (4) What are some good methods for early detection of wheel spinning? and (5) How early can these methods detect wheel spinning with decent performance?

2. DATASET

We used three datasets in our experiments. Two of the datasets were collected from two sections of Algebra content in the MATHia ITS [9] during the 2017-18 school year. Sections within MATHia, which is built on Cognitive Tutor technology from Carnegie Learning, provide instruction and practice on a series of KCs via multi-step problem solving tasks, each problem providing practice on several KCs (we will refer to this as the "CL1 dataset" and "CL2 dataset").

There are 132,551 student-KC pairs in CL1 dataset and 419,832 student-KC pairs in CL2 dataset. The third dataset is from a high school geometry tutor [6] (we will refer to this as the "Geometry dataset") with 8175 student-KC pairs. The datasets used in these experiments were exported from DataShop data [10].

2.1 Label Generation

We labeled each (student, KC) pair in our three datasets according to both the three correct in a row criterion and the PS++ policy [5]. With each criterion, there are three possible label values: *mastered*, *wheel spinning*, and *indeterminate*. A (student, KC) pair was labeled *indeterminate* if there were not enough steps to determine whether the student has or will master the KC. Following [1], we discarded (student, KC) pairs labeled as 'indeterminate' before training our models, given insufficient data to apply the criterion.

2.1.1 Three Correct in a Row

We generated labels for the three correct in a row criterion as follows: For each (student, KC) pair, if there were three or more contiguous correct attempts within the first 10 steps, then the (student, KC) pair was labeled *mastered* (even if there were less than 10 steps). The (student, KC) pairs that did not reach mastery with 10 or more steps were labeled *wheel spinning*. The (student, KC) pairs with less than 10 steps and no occurrence of three contiguous correct steps were labeled *indeterminate*. Under this criterion, the frequency of wheel spinning in CL1, CL2 and Geometry dataset is 6.6%, 0.56%, and 10.2% of (student, KC) pairs, respectively.

2.1.2 Predictive Stability++

The second set of labels are derived from the PS++ policy, which is defined as not reaching a mastery condition *after* a student model's predictions of next step correctness have stabilized to a steady state [5]. In our analysis, we used Bayesian Knowledge Tracing (BKT) as the student model. On the Geometry dataset, we used BKT parameters obtained by fitting the model to data. In the other two datasets, we used the "shipped parameters," that is, the parameters actually used by the ITS. For each step in a (student, KC) pair, BKT calculates $P_C(t)$, which is the probability of getting a correct response for the current step, as well as $P_{C|0}(t) = P(C_{t+1}|\neg C_t)$ and $P_{C|1}(t) = P(C_{t+1}|C_t)$, the probabilities of a correct response on the next step, conditioned on a correct or incorrect response on the current step. When $P_{C|0}(t)$ and $P_{C|1}(t)$ converge, the stopping criterion defined in PS [5] is reached. We then determine the label of the (student, KC) pair as follows: According to PS++ [5], after convergence, when $P_C(t)$ is close enough to its upper bound, we consider the student has mastered this KC. Otherwise the (student, KC) pair is labeled as wheel spinning. For those (student, KC) pairs where the stopping criterion has never been met, we assign *indeterminate* as the label. Under this criterion, the frequency of wheel spinning in CL1, CL2 and Geometry dataset is 24.2%, 2.17%, and 13.2% of (student, KC) pairs, respectively.

2.2 Features

As we explored ways of creating early detectors for wheel spinning, we used a total of 28 features. Among these, 15 were introduced by [3]. These 15 features are extracted to analyze and record three aspects of students' learning progress. The first aspect is students' learning performance like 'correct response count' and 'prior problem count with hint request', which indicate whether the student is doing well on a particular KC. The second aspect is the 'seriousness' of students, including 'prior problem fast correct' and 'prior problem slow incorrect'. These features indicate whether the student appears to be making a deliberate effort on a particular KC. The third category of the features includes general features like 'skill id'. In addition, we used 7 features introduced by [4], and 6 new features based on our previous research and our explorations on the Carnegie Learning dataset. A complete list of features and their descriptions can be found in the online appendix.[1]

3. EXPERIMENTS AND DISCUSSION

We conducted the following experiments and analyses to answer the research questions listed in section 1.

3.1 Compatibility of Operationalizations

3.1.1 Comparing Operationalizations

Regarding Research Question (1) (to what extent do different operationalizations of wheel spinning agree or disagree?), a first observation is that the overall frequency of wheel spinning, reported above, differs substantially under the two operationalizations, with no clear pattern of one predicting more wheel spinning than the other. The confusion matrices (Figure 1, 2, 3) that compare the two operationalizations on each of the three datasets provide further insight into this divergence. For instance, in the CL1 dataset, among all (student, kc) pairs that are labeled as wheel spinning by either operationalization, the two operationalizations agree on only 22.2% of them. In CL2 dataset, the same wheel spinning agreement percentage is 14.1%. In the Geometry dataset, it is 41.6%. The agreement on wheel spinning is generally less than 50%.

	Predictive Stability		
Three Correct in a Row	Master	Indeterminate	Wheel Spinning
Master	31.15%	13.88%	10.01%
Indeterminate	0.23%	29.51%	8.60%
Wheel Spinning	0.32%	0.69%	5.61%

Figure 1: Comparison of two different criteria for wheel spinning in the CL1 dataset.

	Predictive Stability		
Three Correct in a Row	Master	Indeterminate	Wheel Spinning
Master	84.84%	2.35%	1.43%
Indeterminate	0.95%	9.48%	0.40%
Wheel Spinning	0.12%	0.10%	0.34%

Figure 2: Comparison of two different criteria for wheel spinning in the CL2 dataset.

[1]https://tinyurl.com/edm19supplement

Three Correct in a Row	Predictive Stability: Master	Indeterminate	Wheel Spinning
Master	55.96%	7.14%	1.24%
Indeterminate	10.26%	10.12%	5.05%
Wheel Spinning	1.31%	2.06%	6.86%

Figure 3: Comparison of two different criteria for wheel spinning in the Geometry dataset.

To investigate the divergence between the two operationalizations in more detail, we present examples of (student, KC) pairs where the criteria disagree (see Figures 4 and 5). These visualizations show the student first attempted response on each step together with different wheel spinning metrics. Specifically, the student's correct answers, incorrect answers, and hints for the given KC are visualized at the bottom of each graph. When there are 3 contiguous green dots within the first 10 steps (shown in a dash-lined box), the KC will be considered mastered under the three correct in a row criterion. Shown above are $P_C(t)$,$P_{C|0}(t)$,$P_{C|1}(t)$, P_{master}, and upper bound of $P_C(t)$. A vertical, dashed line shows the stopping step for PS; the x-axis denotes the practice opportunity using a 0-based index. In most of the cases in which the three correct in a row criterion and PS++ agree with each other, there is a clear pattern in students' responses towards mastery or wheel spinning. For example, many contiguous corrects will generally result in mastery under both criteria; many contiguous incorrect attempts will result in agreement for wheel spinning. Figure 4 shows one case where PS++ detects wheel spinning but the three correct in a row criterion detects mastery. For this specific (student, KC) pair, mastery (under both criteria) occurs past the point where PS stopped; the initial string of incorrect responses appears to have been influential.

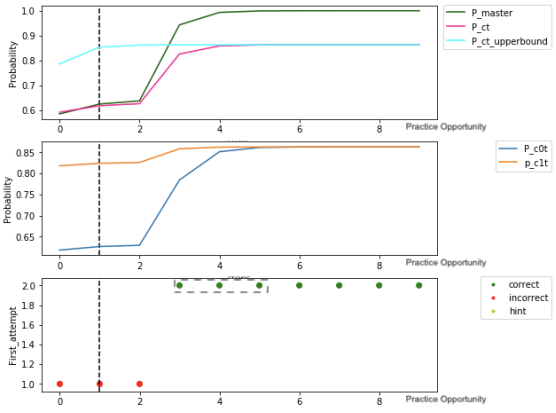

Figure 4: An example of (student, KC) where the three correct in a row criterion detects mastery and PS++ detects wheel spinning.

Figure 5 shows the opposite situation, where PS++ gives a mastery label but the three correct in a row criterion is detecting wheel spinning. In this instance, mastery under the various criteria happens past the 10 step cutoff.

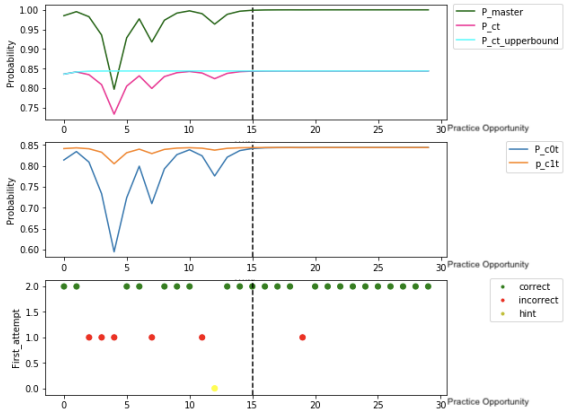

Figure 5: An example of (student, KC) where the three correct in a row criterion detects wheel spinning and PS++ detects mastery.

3.1.2 *Discussion*

We found some overlap but also substantial disagreement between the two operationalizations of wheel spinning, the three correct in a row criterion and PS++. The operationalizations tend to agree when the student's performance is obvious and steady (e.g. the student is doing extremely well or poorly on a KC, or when, as is common, there is a gradual increase in performance). However, these criteria can disagree when students' responses fluctuate. One of the reasons is that the two operationalizations judge mastery in different ways. The three correct in a row criterion, with the explicit 10-step (or other configurable number of steps) cutoff, focuses on mastery in the early stage. Student performance after the cutoff is not taken into account. In contrast, PS++ may consider long-term performance; its mastery judgment can be made using more data, although, as seen in one of our examples, PS++ may stop too early on KCs with lower P_{learn} and an early string of incorrect responses.

3.2 Feature Effectiveness

In this section, we aim to explore the effectiveness of features we are using. In particular, we focus on Research Questions (2): Which set of features lead to better predictions; and (3): What is the simplest set of features that can be effective?

3.2.1 *Feature Importance with Random Forest*

In order to find a set of features for better prediction, we trained a Random Forest model and generated the feature importance graph. Feature importance of a Random Forest is measured by the total decrease in Gini impurity averaged over all the trees in the ensemble [2]. We rely on [8]'s implementation of Random Forest and feature importance and used the default hyperparameters. Figure 6 is an example of the feature importance graph. Among the set of 28 features, four are repeatedly identified as important by the Random Forest model. In all 6 scenarios (three datasets with two wheel spinning criteria), at least three of the four selected features appear in the top five most important features. Two of these features are related to students' performance: 'Correct Response Count' and 'Correct Response Percentage'. The other two features are related to the speed and attentiveness of students: 'Exp Mean Response Time Z-Score'

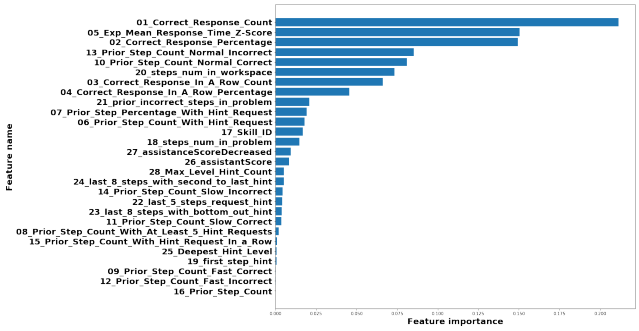

Figure 6: An example of feature importance, trained on CL1 dataset under PS++ criterion.

and 'Prior Step Count Normal Correct'.

3.2.2 *Fitting Single-feature Logistic Regression*

The predictive power of very simple wheel spinning detectors is under-explored in prior literature. Here, we are interested in finding out the predictive value of individual features. Therefore, we picked 6 features that were repeatedly deemed as important in the work reported in section 3.2.1 and built 6 Logistic Regression models, each trained on one of the 6 features. We used [8]'s implementation of Logistic Regression and applied the default hyperparameters.

The results show that the detectors trained on 'Correct Response Count', 'Correct Response Percentage', 'Correct Response in a Row Count' and 'Prior Step Count Normal Correct' achieve high precision and recall when predicting the PS++ label. For example, a Logistic Regression model, trained with 'Correct Response Percentage' as the single feature, detected wheel spinning with 93.5% precision and 77.1% recall after just 4 steps. Generally, features involving correctness seem to be highly effective in wheel spinning detection. In addition, although the detector trained on 'Assistance Score' (i.e. the sum of the number of errors and the number of hints on a step) somewhat surprisingly didn't perform as well as the rest, it still reached 68.4% precision and 60% recall in the fourth step.

3.2.3 *Discussion*

In our experiment, we found that the features that involve correctness of steps tend to be effective in predicting wheel spinning, independent of the criteria or datasets used. These results make sense because intuitively, if a student can get a large percentage of steps correct, then this student is likely on their way to mastering the given KC. They also lead to the question of whether we can build an effective detector that relies on correctness only. The results above show that a Logistic Regression model trained with 'Correct Response Percentage' is able to give us comparable result to other models, although it suffers more from the cold start problem and fluctuates more than other detectors. In addition, other aspects of the step-solving process, including time and help requested, can also be useful, as 'Exp Mean Response Time Z-Score' and 'Assistance Score' also have high feature importance while training Random Forest model. These findings indicate that certain aspects in students' learning performance help predict wheel spinning regardless of the problem setting and tutoring system.

3.3 Early Detection Models

To answer Research Question (4) and (5), we trained multiple machine learning models to study their performance on early detection. We used a Logistic Regression model, trained on the same set of features as in [1], as a baseline. Another detector based on Logistic Regression was trained with the full set of 28 features. In addition, we include one of the detectors used in section 3.2.2, namely, a Logistic Regression model trained on 'Correct Response Percentage', to compare the performance of a simple model with that of other more complex ones. Inspired by [4], we also included a Random Forest model. Finally, we trained two neural-network-based detectors: a 5-layer fully-connected artificial neural network and a 3-layer LSTM. We split our dataset into training and testing data with a 6:4 ratio.

In order to study how early the detectors could accurately detect wheel spinning - by early we mean early in the opportunity count for any given (student, KC) pair - we fitted Random Forest, Logistic Regression and MLP models separately for each practice opportunity - that is, separately with data up to and including opportunity N, for N from 1 to the available data for the given (student, KC) pair. (The labels were computed based on all data, as described above.) Doing so was necessary for these three models as they do not have a recurrent structure to handle variable length steps and most of our features are accumulative. By contrast LSTMs are inherently recurrent, so we trained it on data from every step. We rely on [8]'s implementation of Logistic Regression, Random Forest and Multi-Layer Perceptron (MLP) and [7]'s implementation of LSTM. For Logistic Regression and Random Forest, we used the default hyperparameter provided by [8]. For MLP, we had 3 hidden layers with 64, 32, 16 units, respectively. For LSTM, we used 2 layers with 64 hidden units. For each dataset, we evaluated our detectors on the corresponding test data, and computed precision and recall for the wheel spinning class. Figure 7 shows the detectors' performance on early detection, with two (precision, recall) plot pairs for each detector on 'wheel spinning' class, one pair for each of the two wheel spinning operationalizations.

3.3.1 *Model Performance*

We found that these detectors in general perform well under the PS++ criterion. Most of them reached more than 60% precision and recall after the fourth step, and more than 80% precision and accuracy after the sixth step. Under three correct in a row criterion, the detectors in general perform worse in terms of both precision and recall compared to PS++ criterion. In particular, in CL2 dataset, we observed extremely low recall using all the models. As foreshadowed in section 3.2.2, the single-feature Logistic Regression model (blue line in Figure 7) achieved decent performance compared to other more complex models, except for CL2 dataset. Although it tends to have lower precision and recall in earlier steps, after step 4, its performance improves and is comparable to those of Logistic Regression trained on 15 features and Logistic Regression trained on full set of features.

The two most accurate models are Random Forest (red line in Figure 7) and MLP (purple line in Figure 7). In par-

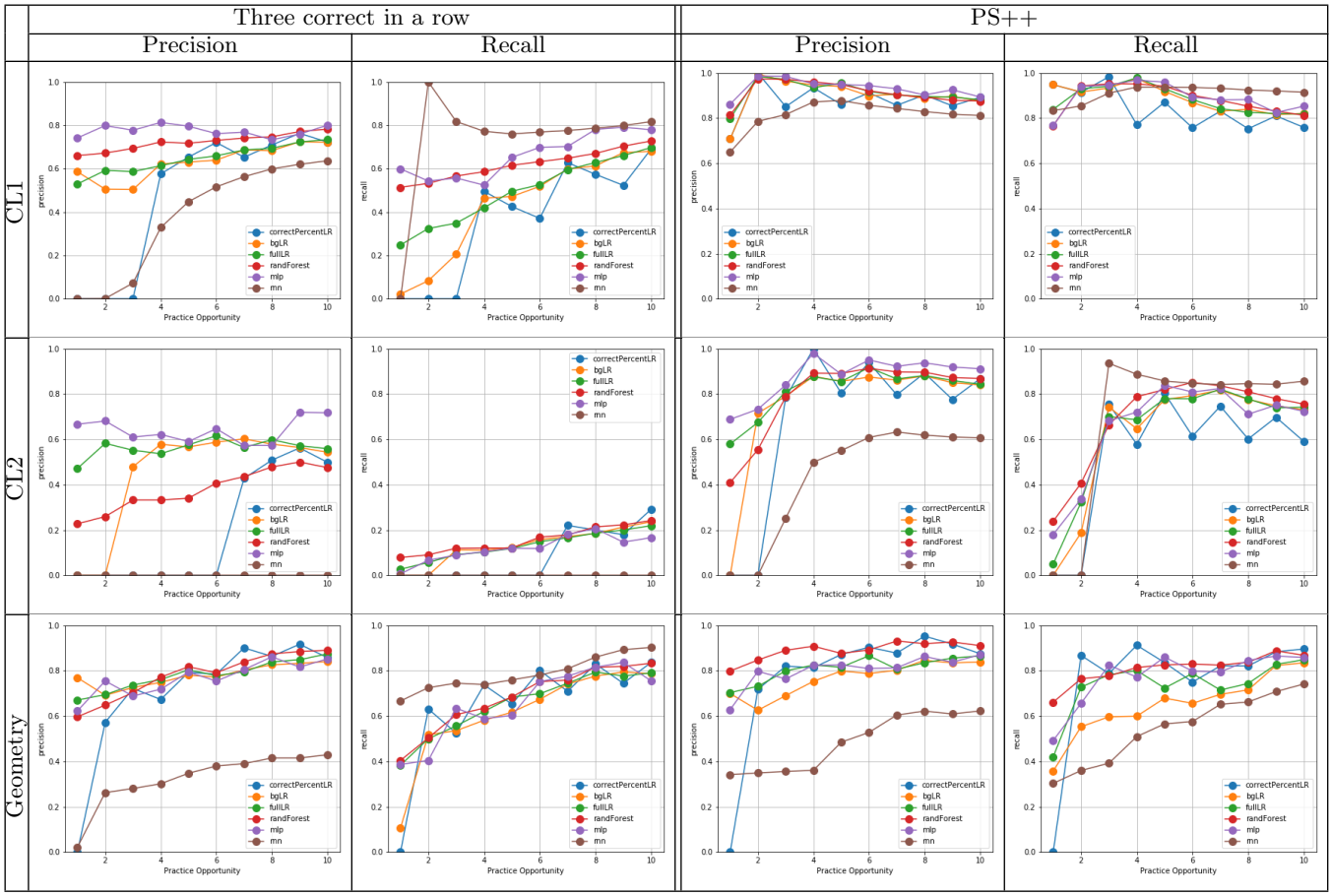

Figure 7: Detectors' early detection Precision and Recall on 'wheel spinning' class in testing data.

ticular, at the fourth step, Random Forest exhibits (77.2%, 63.5%) precision and recall on three correct in a row and (90.8%, 81.4%) on PS++. MLP obtained (71.9%, 58.9%) precision and recall on three correct in a row and (82.4%, 77.2%) on PS++. In comparison, under the same settings, the baseline model achieves (74.9%, 58.2%) and (75.3%, 59.9%). To support comparison, we kept the hyperparameters consistent across all situations. However, if one is interested in getting even better performance, we recommend further tuning these parameters based on specific situation.

3.3.2 Discussion

The models differed in how well they generalize across datasets and wheel spinning criteria. Among them, Random Forest and MLP are those with the most competitive results. Besides its performance, Random Forest also provides us with feature importance evaluation (as in section 3.2.1) and possibly greater interpretability. In addition, to our surprise, we found that the detector trained on a single feature related to correctness consistently produces results not far behind from other detectors. Features unrelated to correctness, on the other hand, are more dependent on the context, such as the different usage of different ITSs and difficulty of the underlying domain. However, the downside of using a single correctness-related feature is that correctness only measures one (albeit important) aspect of the students' behavior while solving steps. Detectors like this may overlook the fact that some students can learn from solving the steps and reach mastery slightly later (e.g. getting the first four steps wrong but answering correctly on the next four) in the process. Nonetheless, this shortcoming should not undermine the power of this model. We recommend its use as a baseline model in future research.

We also find that the detectors perform worse in general under three correct in a row criterion. Even in CL2, where over 97% (student, KC) pairs are under mastery class after discarding the indeterminate class, their performance are extremely low under three correct in a row criterion while decent under PS++ criterion. We failed to come up with a definite explanation to such a phenomenon, but we hypothesize that three correct in a row criterion may not need complicated features to predict, so adding new features merely introduces noise for detectors. Another common issue we found is that these models suffer from the "cold start" problem in every dataset, model, and operationalization. Almost all models had lower than 20% precision and recall in the first three steps in all datasets and metrics on the three correct in a row criterion. This is understandable, since at the first two or three steps, the features collected are often insufficient to determine whether a student has reached mastery or is wheel spinning.

4. CONCLUSION AND FUTURE WORK

In the current work, we aim to move toward clarity and unity in investigations of early detection of wheel spinning. Prior investigations have not compared across different models for early detection across datasets and operationalizations. To begin addressing this gap, we compared two prominent operationalizations of wheel spinning ([1] and [5]) and compared, across three datasets, the performance of several detectors that were trained with different sets of features. First, the frequency of wheel spinning across the three datasets, 0.56%-10.2% under 3-in-a-row and 2.17%-24.2% under PS++ was in line with what previously studies have reported. For example, in the two datasets used by [3], the wheel spinning ratio was 16% and 6%. Further, we found that two well-known operationalizations of wheel spinning diverge substantially in our three datasets, which is not desirable from either a practical or a theoretical perspective, as we do do not know which one to apply or build on. Some typical cases on which they do not agree include getting more KC opportunities correct after [1]'s threshold and answering several consecutive steps incorrectly early followed by much improved performance on later steps. We also found that our models predict PS++ more accurately than they predict [1]'s criterion, most dramatically in the CL2 dataset, but also in the other two data sets (see section 3.3). This finding is surprising especially if one considers that our feature sets included features engineered for prior detectors of three correct in a row reported in the literature [1, 3, 4]. Therefore, it appears unlikely that the more accurate prediction of PS++ is just an artifact of the particular choice of features. What may play a role is that the three correct in a row criterion, in contrast to PS++, does not allow for the fact that KCs vary in difficulty (i.e., require different numbers of practice opportunities, on average, to reach mastery), as is by now well-established [3]. It would be highly desirable to investigate whether our key findings, the discrepancy of the two criteria and the more accurate prediction of PS++, are replicated in a wider range of datasets.

Further, when we evaluated the predictive value of 28 features, we found there is a set of common features that are effective across different datasets and criteria, namely those having to do with step correctness and assistance gained from the system. These features are essential aspects of the learning process to capture for early detection of wheel spinning. Of the 6 wheel spinning detectors we built, the Random Forest model performed consistently well across datasets and operationalizations. Combined with its ability to evaluate feature importance and its potential for interpretability, we recommend trying Random Forest when developing a wheel spinning detector for a tutoring system. In addition, to our surprise, a Logistic Regression model trained on a single feature 'Correct Response Percentage' achieved results that were not far behind those of more complex models like MLP. Addressing our research question of how early we could predict wheel spinning, our best model was able to make predictions with decent precision and recall as early as step 4, namely, on CL1 dataset. These results compare favorably with the accuracy achieved by prior early detectors for wheel spinning, discussed in the introduction. Note that we are not trying to recommend using the fourth step as a definitive criterion for earliness. We merely note that, as our results show, machine-learned models can do well from that particular step onward.

There are several limitations of our work that could be studied further in future work. For example, it would be worthwhile to continue to study agreement and disagreement of additional operationalizations of wheel spinning (e.g., [6]), given that the two we studied agreed to a lesser extent than expected. In addition, some hyperparameters in PS++ can be tuned for a given system, which may further support comparisons.

5. ACKNOWLEDGEMENTS

This work was supported by IES Grants R305A180301 and R305B150008. The opinions expressed are those of the authors and do not represent the views of IES or the U.S. ED.

6. REFERENCES

[1] J. E. Beck and Y. Gong. Wheel-spinning: Students who fail to master a skill. In *AIED*, 2013.

[2] L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984.

[3] Y. Gong and J. E. Beck. Towards detecting wheel-spinning: Future failure in mastery learning. In *Learning @ Scale*, pages 67–74, New York, NY, USA, 2015. ACM.

[4] S. Kai, M. V. Almeda, R. Baker, C. Heffernan, and N. Heffernan. Decision tree modeling of wheel-spinning and productive persistence in skill builders. *JEDM*, pages 36–71, 2018.

[5] T. Käser, S. Klingler, and M. Gross. When to stop?: Towards universal instructional policies. In *LAK*, pages 289–298, New York, NY, USA, 2016. ACM.

[6] R. Liu and K. R. Koedinger. Towards reliable and valid measurement of individualized student parameters. In *EDM*, pages 135–142, 2017.

[7] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[8] 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. *JMLR*, 12:2825–2830, 2011.

[9] S. Ritter and S. Fancsali. Mathia x: The next generation cognitive tutor. In *EDM*, 2016.

[10] J. Stamper, K. Koedinger, R. S. d Baker, A. Skogsholm, B. Leber, J. Rankin, and S. Demi. Pslc datashop: A data analysis service for the learning science community. In *ITS*, pages 455–455. Springer, 2010.