# Investigating the Robustness of Knowledge Tracing Models in the Presence of Student Conceptual Drift

Morgan P Lee
Worcester
Polytechnic Institute
Worcester, MA, USA
mplee@wpi.edu

Artem Frenk
Worcester
Polytechnic Institute
Worcester, MA, USA
afrenk@wpi.edu

Karish A. Gupta
Worcester
Polytechnic Institute
Worcester, MA, USA
kagupta@wpi.edu

Thinh T. Pham
Worcester
Polytechnic Institute
Worcester, MA, USA
ttpham3@wpi.edu

Ethan Croteau
Worcester
Polytechnic Institute
Worcester, MA, USA
ecroteau@wpi.edu

Neil T. Heffernan
Worcester
Polytechnic Institute
Worcester, MA, USA
nth@wpi.edu

Knowledge Tracing (KT) has been an established problem in the educational data mining field for decades, and it is commonly assumed that the underlying learning process being modeled remains static. Given the ever changing landscape of online learning platforms (OLPs), we investigate how the constructs of concept drift and code decay can impact student behavior within an OLP through testing model performance both within a single academic year and across multiple academic years. Four well-studied KT models were applied to five academic years of data to assess how susceptible KT models are to concept drift. Through our analysis, we find that all four families of KT models can lose accuracy with time, and that the relationship between model complexity and susceptability to concept drift is not as simple as previously theorized. Code used to conduct our analyses is available at https://github.com/ASSISTments-IQP/LongitudinalKnowledgeTracing24/tree/master, and the data at https://osf.io/hvfn9/.

**Keywords:** Knowledge Tracing, Concept Drift, Student Modeling, Detector Rot

## 1. INTRODUCTION

To extract meaning from large amounts of data, one may generally assume that the underlying processes which generate said data are static, or at least relatively stable over long periods of time. One of the core goals of educational data mining (EDM) is, of course, using data to model aspects of students' learning processes, which then in turn can be used to predict, explain, or challenge our current understanding of how students learn in online learning platforms (OLPs). Even in cases where the goal is to better suit individual learners' needs, the methodology remains the same: create a model of student behavior and learning through analyzing previously collected data, identify components of that model which may fail to account for individual differences, and update the model to account for the ways in which particular learners differ. These powerful methodologies have allowed researchers to detect the affective state of students (Calvo and D'Mello, 2010; Botelho et al., 2017), model the procedural acquisition of knowledge (Corbett and Anderson, 1994), predict student success (Kovačić, 2010), and detect problematic or unhelpful student behavior (Baker et al., 2010). Many of these approaches have been practiced for multiple decades by now, meaning multiple generations of learners have had their learning processes studied, aggregated, and modeled. As it currently stands, the goal of providing high-quality, scalable educational software that responds to individual student needs (Shemshack and Spector, 2020) is closer than ever before.

As the use of student modeling techniques becomes ever more ubiquitous, it is necessary to revisit the assumptions that guide our practice. We *assume* that data collected from different learners represents the same underlying learning process. We *assume* that the ways in which students learn that are measurable by scientists and practitioners remain consistent. Given the maturation of EDM as a field and the availability of learner data spanning generations of learners, perhaps it is now possible to verify that our assumptions are correct, or at least to identify the circumstances where they are safe assumptions to make.

The educational best-practices of 30 years ago are obviously not the educational best-practices of the modern day. Educational policy has shifted towards meticulous measurement of student progress (Graham and Neu, 2004), identifying failing schools (Nicolaidou and Ainscow, 2005), and standardizing subject curricula to better facilitate rigorous measurement (Popkewitz, 2004). Simultaneously, OLPs rose in popularity, automating student practice and proliferating student engagement data (Ritter et al., 2007; Heffernan and Heffernan, 2014). These educational platforms have also matured since their creation, and every pedagogical and cosmetic change to these platforms could impact the way students interact with these platforms. Even ignoring educational policy changes, students are individuals in a large and changing world, and world events which change how humans relate to one another impact students as much as anyone else. In a particularly extreme example, an entire generation of students experienced learning losses due to the COVID-19 pandemic (Donnelly and Patrinos, 2022). In a changing world, how can we be sure our modeling techniques are still valid?

More specifically, we wish to understand how Knowledge Tracing (KT) models are impacted by changes in student populations over time. KT is a foundational problem of the educational data mining field, and as such we intend to investigate how a number of different modeling techniques behave when applied outside of their temporal context. To achieve this, we draw from Computer Science literature the constructs of code decay and concept drift, and discuss their applicability to online learning platforms. We then propose a methodology for evaluating KT models both within their temporal context and across student populations, taking steps to ensure that our datasets contain similar exercise banks and Knowledge Concepts. We then apply this methodology to four well-studied KT models and examine how each model performs outside of its temporal context. We then conclude by discussing the implications of our findings, as well as limitations and different directions future work could take to overcome said limitations.

Our analysis was guided by the following research questions:

**RQ1.** How robust are KT models to changing student populations?

**RQ2.** Does the complexity of a KT model impact its susceptibility to concept drift?

# 2. BACKGROUND

In this section, we introduce and discuss literature relevant to our investigation of KT model robustness. First, we discuss frameworks for analyzing the drift of software systems from their original contexts (Section 2.1). Next, we introduce KT as a specific machine learning task (Section 2.2) and discuss specific models which will be investigated in this work (Sections 2.2.1-2.2.4). Finally, we discuss relevent prior work investigating the generalizability of KT models (Section 2.3).

## 2.1. SYSTEMIC DECAY

### 2.1.1. Code Decay

Though software clearly cannot decay in the *physical* sense, the contexts in which software is deployed often change from initial design specifications. Thus, code bases must be periodically updated to introduce new features, patch security vulnerabilities, and ensure compatibility with newer hardware. Moreover, the complexity and risk of updating a system to new specifications increases dramatically with the age of the system (Belady and Lehman, 1976). This phenomenon of *code decay* or *code rot* has been part of Computer Science literature for multiple decades, including empirical analyses aiming to classify system components by levels of risk (Ohlsson et al., 1999, for example) and the design of indices measuring the symptoms, risk factors, and predictors of code decay (Eick et al., 2001).

Online learning platforms are software systems like any other, and are thus susceptible to code decay. OLPs must be regularly updated for many of the same reasons as any large-scale software system, and changes in OLPs (such as the introduction of new features or intensifying hardware requirements) have unclear impacts on student behavior within an OLP, potentially introducing noise into data used to train KT models.

### 2.1.2. Concept Drift

Code decay is only one possible source of statistical noise. Others could include environmental factors, differences in teaching practices, or language barriers. The study of *concept drift* aims to detect and correct for this, since any learning process which can be modeled with a distribution is sensitive to changes in that underlying distribution over extended periods of time. Prior works have created methods to detect (Klinkenberg and Joachims, ), explain (Wang et al., 2019), and adapt to (Madireddy et al., 2019) concept drift. More recently, researchers have investigated how concept drift can impact common educational data mining (EDM) and learning analytics (LA) models. Levin et al. (2022) explores how concept drift affects a variety of gaming detectors, finding that contemporary gaming detectors had more trouble generalizing to newer data than classic decision tree based methods, while Deho et al. (2024) found that dataset drift in learning analytics models is linked to algorithmic bias. These works highlight two distinct ways of attempting to quantify concept drift: through longitudinal model evaluation and through the application of concept drift detectors to log data.

## 2.2. KNOWLEDGE TRACING

Long established in EDM literature, Knowledge Tracing (KT) is defined as a many-to-many time-series binary classification problem attempting to predict the correctness of future student responses based on prior performance. Numerous machine learning architectures have been applied to this task, including Factorization Machines (Vie and Kashima, 2019) and psychometric models like Item Response Theory (Yeung, 2019), and Shen et al. (2024) provides a comprehensive survey of historical and contemporary methods. In this work, we will be replicatig four well-studied KT models: Bayesian Knowledge Tracing, Performance Factors Analysis, Deep Knowledge Tracing, and Self Attentive Knowledge Tracing.

### 2.2.1. Bayesian Knowledge Tracing

Originally proposed by Corbett and Anderson (1994) and deeply connected to mastery learning (Bloom, 1968), Bayesian Knowledge Tracing (BKT) models the acquisition of knowledge as a latent Markov process. The student's knowledge state is modeled as a latent variable that is noisily observed through performance on exercises in an intelligent tutoring system. Exercises in said tutoring system are tagged with Knowledge Components (KCs) signifying related items, and items tagged with the same KC are treated as having uniform difficulty. Mastery of different KCs is computed independently, meaning that mastery of one KC is completely independent of other KCs. Due to the assumptions made about students' learning processes, each parameter of a BKT model has a direct interpretation that is explainable to teachers and other educational practitioners (Williamson and Kizilcec, 2021). The latent learning process was originally modeled as one-way, modeling students as unable to forget KCs once they have been mastered, but later model variants explore forgetting behavior (Qiu et al., 2011), individual estimations of prior knowledge (Yudelson et al., 2013), and contextual guess and slip parameters (Baker et al., 2008).

### 2.2.2. Performance Factors Analysis

Closely related to Learning Factors Analysis (Cen et al., 2006), Performance Factors Analysis (PFA) was first proposed in Pavlik et al. (2009) as a logistic regression based alternative to traditional Knowledge Tracing models. Rather than sequentially modeling a student's learning process and updating mastery estimates based on individual student exercises, PFA instead considers the number of correct exercises (wins) and incorrect exercises (fails) by a student on a given KC, along with a KC-level intercept to account for the relative difficulty of a KC. While BKT models KCs as independent entities, PFA can easily predict future performance while accounting for student knowledge of multiple KCs. More recent evaluations of PFA found it to be competitive with more contemporary KT models in certain scenarios (Gervet et al., 2020; Chu and Jr, 2023).

### 2.2.3. Deep Knowledge Tracing

Piech et al. (2015) represents the first application of deep learning methods to the problem of KT. Broadly speaking, Deep Knowledge Tracing (DKT) is the application of a recurrent neural network (RNN) to sequences of exercise-response pairs to predict a student's ability to correctly answer future exercises. Due to its depth, the exact mechanisms by which DKT models student knowledge are less clear than with BKT or PFA. Though deep learning methods can provide performance gains over classical, ”shallower” methods, Khajah et al. (2016) achieve similar performance to the original DKT paper by analyzing certain advantages DKT has over BKT and extending BKT while keeping the underlying model and assumptions the same. In turn, later papers improve on DKT's learning gains by incorporating rich side information into the model (Zhang et al., 2017; Wang et al., 2019).

#### 2.2.4. Self Attentive Knowledge Tracing

With the introduction of attention mechanisms to deep learning in Vaswani et al. (2017), time-series classification problems across numerous domains achieved new state-of-the-art methods. KT was no exception to this, with Pandey and Karypis (2019) proposing an attention mechanism for knowledge tracing. Their aptly named Self Attentive Knowledge Tracing (SAKT) surpassed previous models in performance, while simultaneously showing great promise as more interpretable deep model, given the ability to visualize attention weights to understand particular exercises which the model weighted as more important or explanatory.

### 2.3. Prior Exploration of KT Generalizability

This paper is not the first published work investigating the impact of changing student populations on knowledge tracing models. Lee et al. (2023), which we are directly extending via this journal article, investigated the stability of BKT model predictions over time and found that, while BKT is generally stable year-over-year, large, sudden shifts in student populations can have deleterious effects on model robustness. We wish to replicate these findings by applying BKT to student interaction data spanning more academic years, and also apply the same methodology to other well-known KT models.

## 3. Methods

### 3.1. Data Collection & Preparation

Data for this study was collected using the ASSISTments OLP (Heffernan and Heffernan, 2014), spanning the five academic years between 2019-2020 and 2023-2024. Data not suitable for conducting Knowledge Tracing was filtered out, consisting of all data collected in the months of June, July, and August, as well as problem logs for non-computer-gradable questions, and all problem logs from problem set assigned fewer than 100 times total during the five academic years of interest. Summer student populations often differ greatly to the population of students using an OLP during the school year, while non-computer-gradable problems are incompatible with standard KT models, and removing low-use problem sets from the data lowers the likelihood of models differing solely due to out-of-vocabulary KCs and exercises. Information about the size of the data gathered from each academic year can be found in table 1. The relative size of each year's data is worthy of note. Different years have great differences in the number of available logs, with the largest year having over twenty-one times the amount of total rows. Since the amount of available training data has a large impact on model fitness, this disparity in dataset sizes presents an issue.

To mitigate the impact of our dataset sizes, rather than using all available data for each year, we draw random samples from each available academic year. Randomly sampling *user/exercise interactions* would isolate those rows from their surrounding context, while sampling *per user* reintroduces concerns over differences in training set sizes, as the total number of exercises completed per user varies widely. Instead, we randomly sample 50,000 assignment logs, which are instances of a single student completing an assigned problem set. This allows us to draw

| AY | Total Rows | Assignment Logs | Unique Students | Unique KCs | % Correct |
|---|---|---|---|---|---|
| 2019-2020 | 17,962,663 | 1,645,060 | 228,207 | 408 | 0.728 |
| 2020-2021 | 69,760,692 | 5,478,914 | 437,500 | 411 | 0.697 |
| 2021-2022 | 11,421,033 | 1,309,773 | 122,397 | 412 | 0.686 |
| 2022-2023 | 5,382,200 | 754,299 | 71,284 | 407 | 0.668 |
| 2023-2024 | 3,254,928 | 519,700 | 50,896 | 408 | 0.660 |

Table 1: Dataset sizes after filtering out ineligible problem logs.

samples of consistent size, since problem set length is more consistent, while collecting coherent sequences of student/exercise interactions in their full context. Our final dataset consists of ten samples per academic year, with each sample containing 50,000 assignment logs each[1].

### 3.2. Study Design

In order to effectively investigate the susceptibility of KT models to concept drift, we need to establish baseline performance for each model on each target year and somehow evaluate models in a cross-year context. To measure within-year performance, we conducted a five-fold cross validation, with each fold consisting of two samples, gathering five AUC measurements per model per year. To investigate model performance across years, for each sample of a target year, we trained a model on the full sample and evaluated the fit model on one sample from all *subsequent* years. While it's clearly possible to evaluate a model using data gathered *before* the training year, doing so is more of an analytical tool, as in real systems possibly affected by concept drift, model accuracy decreases due to the introduction of *later* data. Thus, we only evaluate models using data from their training year or later. This gathers ten AUC measurements per model per evaluation year.

## 4. Results

Each model was implemented in Python 3.12[2] largely following the methods of their original papers, with the following differences. BKT was implemented with the forgetting parameter enabled via the hmmlearn package. After fitting models for each available KC in the training set, learned parameters were averaged to make a ”best guess” KT model in the case of evaluating KCs that were not present in the training set. PFA was implemented using scikit-learn, fitting separate covariates for wins and fails for each KC, along with a KC level intercept and parameters for KCs not present in the training set. DKT was implemented in keras with a tensorflow backend, consisting of an embedding layer, followed by an LSTM with 124 hidden units and a batch size of 64, training each model for five epochs. Rather than using exercise labels for DKT model inputs, we use KC tags as model inputs due to the large number of exercises present in each dataset. SAKT was also implemented in keras with a tensorflow backend, using labels as inputs, following the architecture proposed by Pandey and Karypis (2019), with a batch size of 16, and fitting for 5 epochs in the cross-year case and one epoch in the within-year case (as

[1] These samples are available at https://osf.io/hvfn9/

[2] These implementations, along with analysis code, are available at https://github.com/ASSISTments-IQP/LongitudinalKnowledgeTracing24

within-year training has more training data). Both DKT and SAKT were fit using the ADAM optimizer with a learn rate of 0.001 on an NVIDIA A100 GPU.

Model evaluation results can be found in figure 1, organized by evaluation year. Different bar colors indicate the different years used to train models before evaluation on the target year. Perhaps unsurprisingly, models generally performed better when evaluated on their training year compared to other years, with the notable exception of many DKT models. The difference between train and evaluation year performance is much more pronounced with SAKT compared to other models.

Finally, to investigate the significance of model performance dropoff, we performed statistical tests for all models on all target years except 2019-2020 (there is no comparison to be made for this year, since only within-year analysis was conducted for this year). Depending on the number of model training years available for an evaluation year, we performed either a Welch's t test or a one-way ANOVA. The results of those tests can be found in table 2. All models evaluated on all years featured significant differences in model performance with the exception of PFA and DKT on 21-22 and PFA only on 20-21.

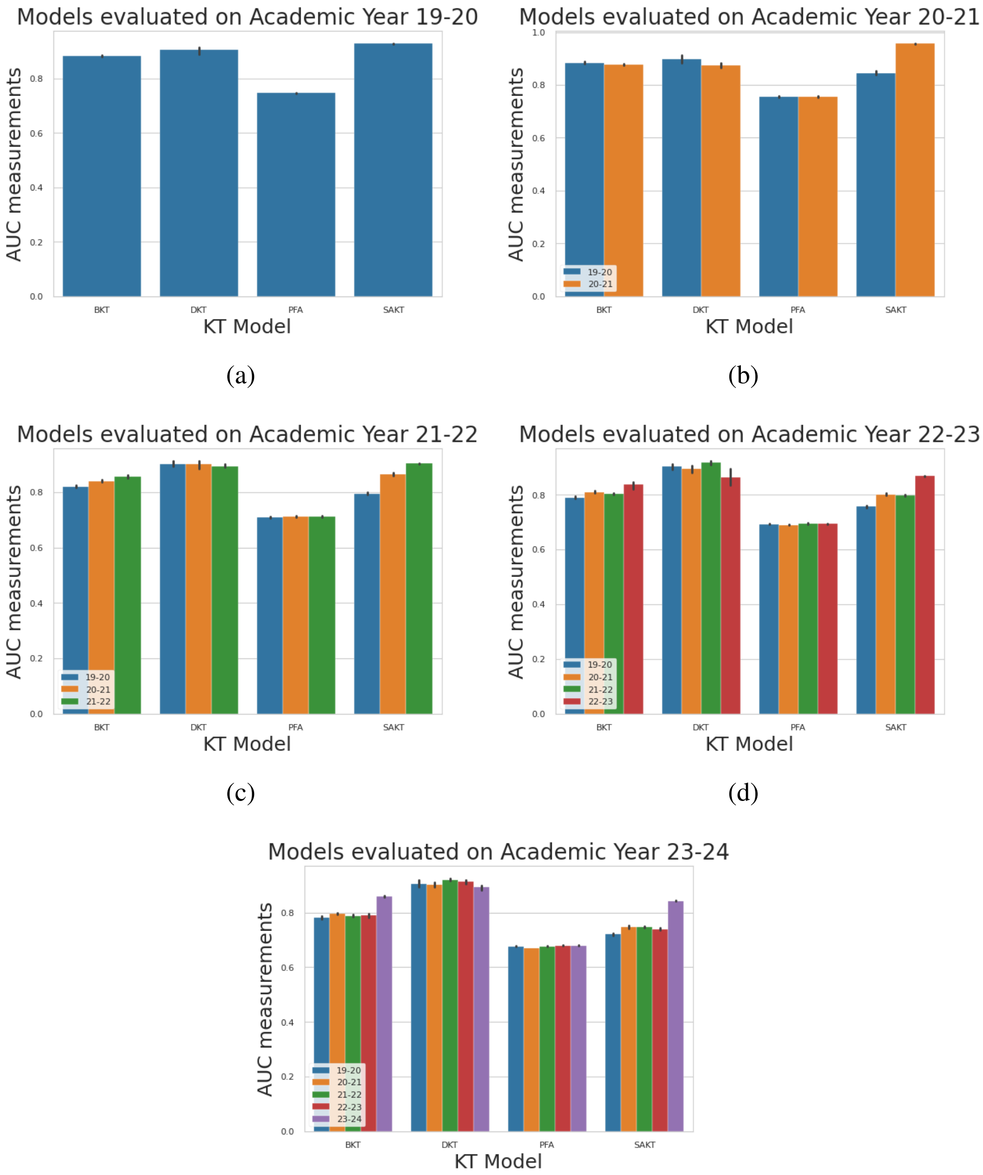
| Model | Evaluation Year | Test Statistic | p |
|---|---|---|---|
| BKT | 20-21 | t = 2.822 | 0.014** |
| | 21-22 | F = 57.05 | <0.001*** |
| | 22-23 | F = 33.35 | <0.001*** |
| | 23-24 | F = 75.05 | <0.001*** |
| PFA | 20-21 | t = 0.313 | 0.761 |
| | 21-22 | F = 1.074 | 0.359 |
| | 22-23 | F = 3.116 | 0.040** |
| | 23-24 | F = 22.74 | <0.001*** |
| DKT | 20-21 | t = 2.402 | 0.032** |
| | 21-22 | F = 0.332 | 0.721 |
| | 22-23 | F = 6.954 | 0.001*** |
| | 23-24 | F = 3.605 | 0.013** |
| SAKT | 20-21 | t = -32.70 | <0.001*** |
| | 21-22 | F = 547.6 | <0.001*** |
| | 22-23 | F = 476.8 | <0.001*** |
| | 23-24 | F = 281.2 | <0.001*** |

Table 2: Test results for significant differences when evaluating models across years

# 5. Discussion

## 5.1. RQ1: KT Robustness

Our results suggest that all KT models we tested are vulnerable to concept drift under some conditions. This includes BKT, contradicting prior work from Lee et al. (2023), potentially due to an error in the modeling methodology. Lee et al. (2023) applied BKT to every student *attempt* rather than every exercise, resulting in lower average AUCs than reported in the current work. Given that we filtered our training data to ensure comparability, this indicates there are sources



(a)

(b)

(c)

(d)

(e)

Figure 1: AUC measurements for each model evaluated on each academic year

of concept drift that impact student learning in OLPs beyond statistical noise in which problem sets are assigned.

It is also telling that model performance degradation seems linked to models being used outside of their temporal context. That is, model degradation is more pronounced when evaluated on data far newer than the data used to train the original model.

## 5.2. RQ2: Model Complexity & Robustness

Our initial hypothesis Based on our findings, KT model complexity (as measured by the number of trainable parameters) may not be directly linked to lower model robustness as theorized in Levin et al. (2022). While SAKT exhibited the steepest drop-off in model performance over time, BKT also experienced statistically significant losses in AUC despite having orders of magnitude fewer tunable parameters, and though DKT lies between them in complexity, it appears more robust to concept drift than either BKT or SAKT. As our tests show that PFA models are less sensitive to concept drift, perhaps the difference lies in how each model conceptualizes the learning process. BKT, DKT, and SAKT all model learning by making inferences on a full sequence of student responses, while PFA relies on an aggregated, engineered feature that may limit the impact of noise in student response sequences.

## 6. Limitations & Future Work

While our findings broadly suggest that KT models are susceptible to concept drift, there are notable limitations in our analysis. Our model-focused approach to measuring concept drift cannot describe how the distribution of student responses has changed, nor explain factors which could be causing said change. Clearly *something* about the interactions students have within an OLP changes through time, and future works could employ a data-centric approach to detecting concept drift alongside evaluating models through time. Our model evaluation methodology gathers five within-year measurements of AUC while gathering ten cross-year AUC measurements per evaluation year, resulting in our measurement datasets having differing sizes. Additionally, within-year training is conducted on more data than cross-year data. Further steps could be taken to ensure models are all trained on similar amounts of data to mitigate the impact of dataset size on evaluation results. As noted in section 4, we use KC tags as inputs to our DKT model instead of exercise tags, while using exercise tags for SAKT. It is entirely possible that the choice of using KCs as model inputs instead of exercises could be responsible for DKT's relative robustness. Future work could investigate if training on exercise tags or KC tags impacts model accuracy and robustness. Finally, we only evaluated basic implementations of our four KT models. As discussed in Shen et al. (2024), our four choices of models represent broader ”families” of KT models, and novel KT modeling is an area of active research in EDM. Exploring how extensions to these model families impacts robustness would also give more insight into the relationship between model complexity and generalizability.

## 7. Conclusion

In contrast to previous findings, this study indicates that knowledge tracing models can indeed lose predictive power over time. Models which rely on engineered representations of student performance may be less sensitive to concept drift, but will still lose power over a long enough time frame. These findings indicate that the underlying process of student learning (as monitored through student interaction logs) may not be as stable as previously theorized. Investigations targeting student interaction data could yield further insights into how and why student behavior changes, and may be key in creating models and training schedules that are robust in the presence of concept drift. There are also many more common educational models, such as affect detectors and psychometric models, which may have differing levels of robustness to changing student populations. Understanding which models lose accuracy over time and why is an essential step in understanding how student learning behavior changes over time.

## Acknowledgements

We would like to thank NSF (e.g., 2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (e.g., R305N210049, R305D210031, R305A170137, R305A170243, R305A180401, & R305A120125), GAANN (e.g., P200A180088 & P200A150306), EIR (U411B190024 & S411B210024), ONR (N00014-18-1-2768), NHI (R44GM146483), and Schmidt Futures. None of the opinions expressed here are that of the funders.

## References

Baker, R. S. J. D., Corbett, A. T., and Aleven, V. 2008. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In *Intelligent Tutoring Systems*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, B. P. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, Eds. Vol. 5091. Springer Berlin Heidelberg, Berlin, Heidelberg, 406–415. Series Title: Lecture Notes in Computer Science.

Baker, R. S. J. d., Mitrović, A., and Mathews, M. 2010. Detecting Gaming the System in Constraint-Based Tutors. In *User Modeling, Adaptation, and Personalization*, P. De Bra, A. Kobsa, and D. Chin, Eds. Springer, Berlin, Heidelberg, 267–278.

Belady, L. A. and Lehman, M. M. 1976. A model of large program development. *IBM Systems Journal 15,* 3, 225–252. Conference Name: IBM Systems Journal.

Bloom, B. S. 1968. Learning for Mastery. Instruction and Curriculum. Regional Education Laboratory for the Carolinas and Virginia, Topical Papers and Reprints, Number 1. *Evaluation Comment 1,* 2 (May). Publisher: Regional Education Laboratory for the Carolinas and Virginia, Mutual Plaza (Chapel Hill and Duke Sts.), Durham, N.C. 27701.

Botelho, A. F., Baker, R. S., and Heffernan, N. T. 2017. Improving Sensor-Free Affect Detection Using Deep Learning. In *Artificial Intelligence in Education*, E. André, R. Baker, X. Hu, M. M. T. Rodrigo, and B. du Boulay, Eds. Springer International Publishing, Cham, 40–51.

Calvo, R. A. and D'Mello, S. 2010. Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications. *IEEE Transactions on Affective Computing 1,* 1 (Jan.), 18–37. Conference Name: IEEE Transactions on Affective Computing.

Cen, H., Koedinger, K., and Junker, B. 2006. Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement. In *Intelligent Tutoring Systems*, M. Ikeda, K. D. Ashley, and T.-W. Chan, Eds. Springer, Berlin, Heidelberg, 164–175.

Chu, W. and Jr, P. I. P. 2023. The Predictiveness of PFA is Improved by Incorporating the Learner's Correct Response Time Fluctuation. ISBN: 9781733673648 Pages: 244–250 Publication Title: Proceedings of the 16th International Conference on Educational Data Mining Publisher: International Educational Data Mining Society.

Corbett, A. T. and Anderson, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction 4,* 4 (Dec.), 253–278.

Deho, O. B., Liu, L., Li, J., Liu, J., Zhan, C., and Joksimovic, S. 2024. When the Past != The Future: Assessing the Impact of Dataset Drift on the Fairness of Learning Analytics Models. *IEEE Transactions on Learning Technologies 17*, 1007–1020. Conference Name: IEEE Transactions on Learning Technologies.

DONNELLY, R. AND PATRINOS, H. A. 2022. Learning loss during Covid-19: An early systematic review. *PROSPECTS 51,* 4 (Oct.), 601–609.

EICK, S., GRAVES, T., KARR, A., MARRON, J., AND MOCKUS, A. 2001. Does code decay? Assessing the evidence from change management data. *IEEE Transactions on Software Engineering 27,* 1 (Jan.), 1–12. Conference Name: IEEE Transactions on Software Engineering.

GERVET, T., KOEDINGER, K., SCHNEIDER, J., AND MITCHELL, T. 2020. When is Deep Learning the Best Approach to Knowledge Tracing? *Journal of Educational Data Mining 12,* 3 (Oct.), 31–54. Number: 3.

GRAHAM, C. AND NEU, D. 2004. Standardized testing and the construction of governable persons. *Journal of Curriculum Studies 36,* 3 (May), 295–319. Publisher: Routledge _eprint: https://doi.org/10.1080/0022027032000167080.

HEFFERNAN, N. T. AND HEFFERNAN, C. L. 2014. The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. *International Journal of Artificial Intelligence in Education 24,* 4 (Dec.), 470–497.

KHAJAH, M., LINDSEY, R. V., AND MOZER, M. C. 2016. How deep is knowledge tracing? In *arXiv.org*.

KLINKENBERG, R. AND JOACHIMS, T. Detecting concept drift with support vector machines.

KOVAČIĆ, Z. J. 2010. Early prediction of student success: Mining students' enrolment data. In *Proceedings of Informing Science & IT Education Conference (InSITE).*

LEE, M. P., CROTEAU, E., GURUNG, A., BOTELHO, A. F., AND HEFFERNAN, N. T. 2023. Knowledge Tracing over Time: A Longitudinal Analysis. International Educational Data Mining Society. ERIC Number: ED630851.

LEVIN, N., BAKER, R., NASIAR, N., STEPHEN, F., AND HUTT, S. 2022. Evaluating Gaming Detector Model Robustness Over Time. *Proceedings of the 15th International Conference on Educational Data Mining, International Educational Data Mining Society*.

MADIREDDY, S., BALAPRAKASH, P., CARNS, P., LATHAM, R., LOCKWOOD, G. K., ROSS, R., SNYDER, S., AND WILD, S. M. 2019. Adaptive Learning for Concept Drift in Application Performance Modeling. In *Proceedings of the 48th International Conference on Parallel Processing*. ICPP '19. Association for Computing Machinery, New York, NY, USA, 1–11.

NICOLAIDOU, M. AND AINSCOW, M. 2005. Understanding Failing Schools: Perspectives from the inside. *School Effectiveness and School Improvement 16,* 3 (Sept.), 229–248. Publisher: Routledge _eprint: https://doi.org/10.1080/09243450500113647.

OHLSSON, M., VON MAYRHAUSER, A., MCGUIRE, B., AND WOHLIN, C. 1999. Code decay analysis of legacy software through successive releases. In *1999 IEEE Aerospace Conference. Proceedings (Cat. No.99TH8403)*. Vol. 5. 69–81 vol.5.

PANDEY, S. AND KARYPIS, G. 2019. A Self-Attentive model for Knowledge Tracing. arXiv:1907.06837 [cs, stat].

PAVLIK, P. I., CEN, H., AND KOEDINGER, K. R. 2009. Performance Factors Analysis – A New Alternative to Knowledge Tracing. Tech. rep. Publication Title: Online Submission ERIC Number: ED506305.

PIECH, C., BASSEN, J., HUANG, J., GANGULI, S., SAHAMI, M., GUIBAS, L. J., AND SOHL-DICKSTEIN, J. 2015. Deep Knowledge Tracing. In *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc.

POPKEWITZ, T. S. 2004. Educational Standards: Mapping Who We Are and Are to Become. *Journal of the Learning Sciences 13,* 2 (Apr.), 243–256.

QIU, Y., QI, Y., LU, H., PARDOS, Z., AND HEFFERNAN, N. 2011. Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing. 139–148.

RITTER, S., ANDERSON, J. R., KOEDINGER, K. R., AND CORBETT, A. 2007. Cognitive Tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review 14,* 2 (Apr.), 249–255.

SHEMSHACK, A. AND SPECTOR, J. M. 2020. A systematic literature review of personalized learning terms. *Smart Learning Environments 7,* 1 (Oct.), 33.

SHEN, S., LIU, Q., HUANG, Z., ZHENG, Y., YIN, M., WANG, M., AND CHEN, E. 2024. A Survey of Knowledge Tracing: Models, Variants, and Applications. *IEEE Transactions on Learning Technologies 17*, 1898–1919. Conference Name: IEEE Transactions on Learning Technologies.

VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., AND POLOSUKHIN, I. 2017. Attention Is All You Need. arXiv:1706.03762.

VIE, J.-J. AND KASHIMA, H. 2019. Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing. *Proceedings of the AAAI Conference on Artificial Intelligence 33,* 01 (July), 750–757. Number: 01.

WANG, X., WANG, Z., SHAO, W., JIA, C., AND LI, X. 2019. Explaining Concept Drift of Deep Learning Models. In *Cyberspace Safety and Security*, J. Vaidya, X. Zhang, and J. Li, Eds. Springer International Publishing, Cham, 524–534.

WANG, Z., FENG, X., TANG, J., HUANG, G. Y., AND LIU, Z. 2019. Deep Knowledge Tracing with Side Information. In *Artificial Intelligence in Education*, S. Isotani, E. Millán, A. Ogan, P. Hastings, B. McLaren, and R. Luckin, Eds. Springer International Publishing, Cham, 303–308.

WILLIAMSON, K. AND KIZILCEC, R. F. 2021. Effects of Algorithmic Transparency in Bayesian Knowledge Tracing on Trust and Perceived Accuracy. Tech. rep., International Educational Data Mining Society. ERIC Number: ED615541.

YEUNG, C.-K. 2019. Deep-IRT: Make Deep Learning Based Knowledge Tracing Explainable Using Item Response Theory. arXiv:1904.11738.

YUDELSON, M. V., KOEDINGER, K. R., AND GORDON, G. J. 2013. Individualized Bayesian Knowledge Tracing Models. In *Artificial Intelligence in Education*, H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, Eds. Springer, Berlin, Heidelberg, 171–180.

ZHANG, L., XIONG, X., ZHAO, S., BOTELHO, A., AND HEFFERNAN, N. T. 2017. Incorporating Rich Features into Deep Knowledge Tracing. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*. L@S '17. Association for Computing Machinery, New York, NY, USA, 169–172.