

Investigating the Robustness of Knowledge Tracing Models in the Presence of Student Concept Drift

Morgan Lee ^{*†} Artem Frenk ^{*} Eamon Worden
Worcester Polytechnic Institute Worcester Polytechnic Institute Worcester Polytechnic Institute
Karish Gupta Tinh Pham
Worcester Polytechnic Institute Worcester Polytechnic Institute
Ethan Croteau Neil Heffernan
Worcester Polytechnic Institute Worcester Polytechnic Institute

ABSTRACT

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 concept drift and changing student populations 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 exhibit degraded performance, Bayesian Knowledge Tracing (BKT) remains the most stable KT model when applied to newer data, while more complex, attention based models lose predictive power significantly faster. To foster more longitudinal evaluations of KT models, the data used to conduct our analysis is available at https://osf.io/hvfn9/?view_only=b936c63dfdae4b0b987a2f0d4038f72a.

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. These models can then 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 by 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 [6, 7], model the procedural acquisition of knowledge [10], predict student success [20], and detect problematic or unhelpful student behavior [4, 35]. 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 [33] 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 [14], identifying failing schools [24], and standardizing subject curricula to better facilitate rigorous measurement [30]. Simultaneously, OLPs rose in popularity, automating student practice and proliferating student engagement data [32, 15]. 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

^{*}Denotes equal contribution

[†]Correspondence to mplee@wpi.edu

COVID-19 pandemic [12]. 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 borrow from Data Mining literature the concept of dataset shift, and discuss its 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?

In addition to the listed analyses above, there is a notable lack of knowledge tracing datasets that span multiple years. Alongside this paper we will be releasing a publicly available dataset containing student interaction data spanning multiple academic years, allowing other researchers to evaluate their own models under controlled conditions of "aging."

2. BACKGROUND

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

2.1 Knowledge Tracing

Long established in EDM literature, 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 [37] and psychometric models like Item Response Theory [41]. Shen et al. [34] provides a comprehensive survey of historical and contemporary methods. In this work, we will be replicating four well-studied KT models: Bayesian Knowledge Tracing, Performance Factors Analysis, Deep Knowledge Tracing, and Self-Attentive Knowledge Tracing.

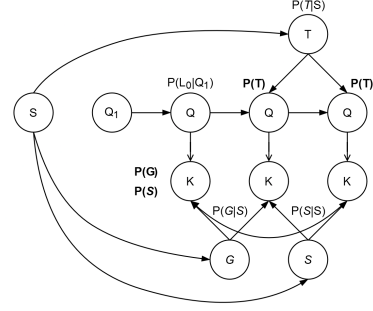


Figure 1: Model architecture for BKT

Bayesian Knowledge Tracing. Originally proposed by Corbett & Anderson [10] and deeply connected to mastery learning [5], 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 [40]. 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 [31], individual estimations of prior knowledge [42], and contextual guess and slip parameters [3].

Performance Factors Analysis. Closely related to Learning Factors Analysis [8], Performance Factors Analysis (PFA) was first proposed in Pavlik et al. [27] 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 [13, 9].

Deep Knowledge Tracing. Piech et al. [29] 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

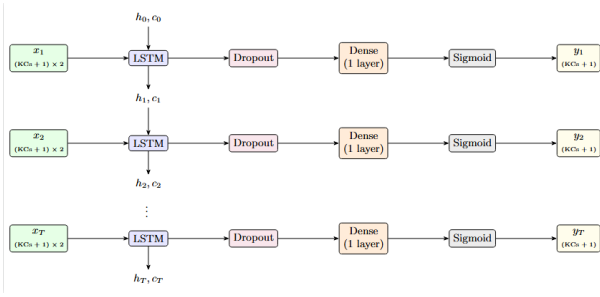


Figure 2: Model architecture for DKT

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. [18] 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 [43, 39].

Self-Attentive Knowledge Tracing. With the introduction of attention mechanisms to deep learning in Vaswani et al. [36], time series classification problems across numerous domains achieved new state-of-the-art methods. KT was no exception to this, with Pandey & Karypis [25] 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 a more interpretable deep model, given the ability to visualize attention weights to understand particular exercises which the model weighted as more important or explanatory.

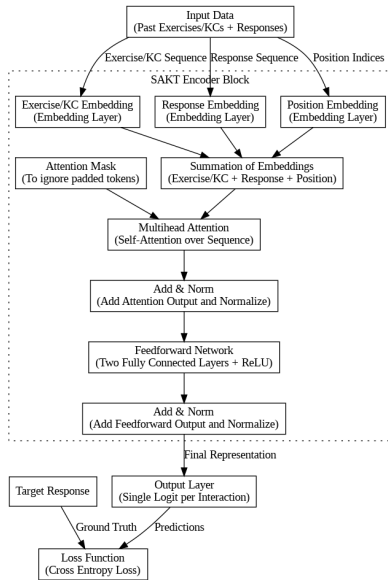


Figure 3: Model architecture for SAKT

2.2 Distributional Shifts

Broadly speaking, for a given supervised learning problem with training set X and labels Y , we are interested in modeling the joint probability distribution:

$$P(X, Y) = P(Y|X)P(X) = P(X|Y)P(Y)$$

Changes in this joint distribution can be categorized based on the part of this distribution that changes [17]:

1. **Covariate Shift**, a change $P(X)$ while $P(Y|X)$ remains the same.
2. **Label Shift**, a change in $P(Y)$ while $P(X|Y)$ remains the same.
3. **Concept Drift**, a change in $P(Y|X)$ while $P(X)$ remains the same.

Concept drift is perhaps the most difficult of these three shifts to adapt to, since a supervised learning model explicitly attempts to estimate $P(Y|X)$. Prior works have created methods to detect [19], explain [38], and adapt to [23] 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. [22] 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. [11] found that concept drift in LA 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.

2.3 Prior Exploration of KT Generalizability

Covariate Shift, Label Shift, and Concept Drift all have the potential to decrease the performance of KT models. Covariate and/or label shift could be introduced by an influx of new students, or a new curriculum being added to an OLP. Since KT models explicitly try to model the acquisition of knowledge, identifying when a model is susceptible to concept drift simultaneously raises questions about the underlying learning process.

This paper is not the first published work investigating the impact of changing student populations on knowledge tracing models. Lee et al. [21] 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 and extend these findings by investigating the performance of other well-known KT models, including BKT, when applied to student interaction data spanning a longer time frame.

3. METHODS

3.1 Data Collection & Preparation

Data for this study was collected using the XXX OLP, 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

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.

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 problem logs. 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 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 such samples¹ per academic year, with samples containing 50,000 assignment logs each².

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 ten-fold cross validation, training one model per sample and evaluating it on the other nine samples³. 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

¹In this paper, “sample” refers specifically to one of these random samples of 50,000 student/assignment interactions, *not* individual examples of model inputs & outputs.

²These samples are available here

³Rather than doing a classic 90/10 train-test split for cross-validation, we opt to train on one sample and evaluate on the other nine to make sure all models are trained on roughly the same amount of data

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. Additionally, to explicitly investigate the effect of overparameterization on model performance over time, two different versions of SAKT were evaluated: one using KCs as model input and one using exercises directly.

3.3 Model Implementations

Each model was implemented in Python 3.12⁴, 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` [28], 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. Both SAKT and DKT were implemented in `pytorch` [26] and trained on NVIDIA A100 GPUs. Visual representations of model architectures can be found in figures 1–3. For a full mathematical explanation of each model, please refer to appendix A.

3.3.1 Hyperparameter Tuning

To tune the hyperparameters of our deep models, we used Bayesian hyperparameter optimization. Hyperparameters tuned for each model can be found in table 2. Rather than tuning these hyperparameters with the samples constructed for our main study, we constructed a new validation sample by sampling 100,000 unique assignment logs from all available academic years. Using the `optuna` package [1], we performed a four-fold cross-validation using the validation sample for each sampled hyperparameter combination, terminating after fifty rounds of cross-validation were performed.

4. RESULTS

Model evaluation results can be found in figure 4 which shows the AUC of each model compared to the years since the model was trained organized by model. Every model tested by our method had decreasing AUC over time. All five models have significant correlations between model fit metrics and the number of years between training & evaluation data (see table 4 for estimated correlation coefficients) indicating that increased age since training worsens model performance. BKT is notable for this relationship not being strictly monotonic. This suggests that after 3 or 4 years

⁴These implementations, along with analysis code, are available here

Parameter	Range	DKT	SAKT-E	SAKT-KC
num_steps	[20,100]	40	60	100
batch_size	[16,64]	16	64	48
d_model	[64,512]	96	352	128
num_epochs	DKT [100, 300] SAKT [10, 40]	100	23	25
dropout_rate	[0.1, 0.5]	0.278	0.47	0.188
learn_rate	[1e-4, 1e-2]	2e-3	1e-4	1e-4
reg_lambda	[1e-6, 1e-2]	1.4e-5	n/a	n/a
num_heads	[2,4,8,16,32]	n/a	8	16
learn_decay_rate	[0.7, 0.99]	n/a	0.7	0.868

Table 2: Learned Hyperparameters for DKT, SAKT-E, and SAKT-KC

since training, the decrease in performance may have hit a maximum and they may continue to perform similarly while the other models will take more continue to have decreasing performance past 4 years. BKT and PFA lost the least predictive power across time, losing only ≈ 0.03 & ≈ 0.05 AUC, while SAKT-E was the most impacted, losing ≈ 0.17 AUC.

Figure 5 shows the log loss of each model compared to the number of years since the models were trained. All models had increasing log loss as the number of years increased, with SAKT-E having the greatest increase of ≈ 0.3 and DKT the smallest of ≈ 0.05 . SAKT-E and SAKT-KT both continue to decrease in performance across all years, while PFA, DKT and BKT stop worsening around the final year. This continues to suggest that BKT, DKT and possibly PFA, which had the smallest drop in performance for the AUC model, may have reached their worst performance and continue to have a steady log loss. The final figure of F1 Score and years since training shows similar results, with SAKT-E and SAKT-KC decreasing the most and BKT DKT and PFA decreasing less, and leveling out towards the end.

Figure 6 shows the F1 score of each model compared to the number of years since the models were trained. Unlike with AUC, *every* model’s performance was strictly decreasing. Similar to the trends in AUC and Log Loss, the two SAKT models showed the sharpest declines, with SAKT-KC’s decline looking broadly linear. SAKT-E experiences a sharper decrease between zero and one years between training & evaluation, then appears to decline at a similar rate to SAKT-KC.

Finally, to compare the rates of model performance loss, we performed the following fixed effects regression on all three reported metrics:

$$\text{metric} = \alpha_m + \beta y + \beta_m y$$

Where α_m is the estimated fixed effect for model m , y is the number of years between training and evaluation data, β is our estimated rate of performance loss for the reference category and β_m is the interaction term between α_m and β . Results from these three regressions can be found in tables 5–7.

5. DISCUSSION

Our results suggest that all KT models we tested are vulnerable to concept drift under some conditions. This includes

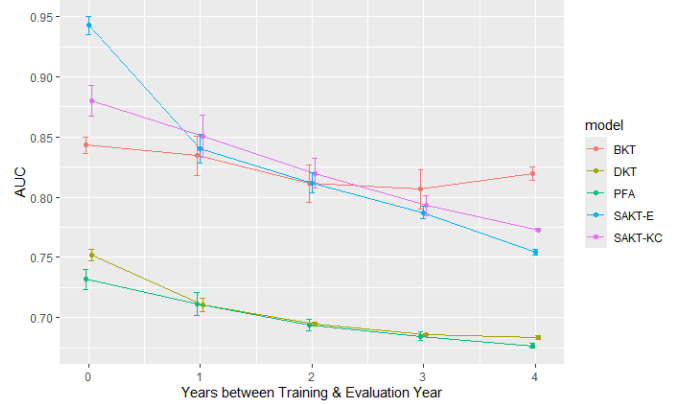


Figure 4: Mean AUC measurements vs. training data age. Error bars represent a 95% CI for given training data age.

Model	Proposed	Logical Unit	Tunable Params
BKT	[10]	KC	2,012
PFA	[27]	KC	1,208
DKT	[29]	KC	734,026
SAKT-KC	[25]	KC	262,982
SAKT-E	[25]	Exercise	12,400,658

Table 3: Number of trainable parameters of each model.

BKT, though BKT seems the most reliably robust by all three recorded metrics. Every other model we assessed had more extreme rates of performance loss (see the interaction terms in tables 5–7). 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. When evaluated on data from four years later, every model examined lost at *least* 0.05 AUC. However, for BKT, DKT, and PFA the performance degradation appears to slow down if not completely halt after a certain time frame. This suggests that there may be a certain number of years after which the maximum model degradation has occurred and model’s retain their predictive power year after year. For BKT which had a final AUC of 0.82 this would indicate that it may be viable to use well past the year of its training data. Similarly, DKT and PFA may continue to be used past the years on which they were trained without expecting serious amounts

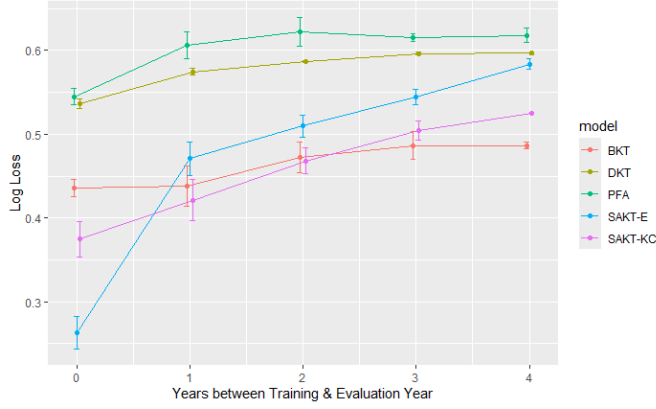


Figure 5: Mean Log Loss measurements vs. training data age. Error bars represent a 95% CI for given training data age.

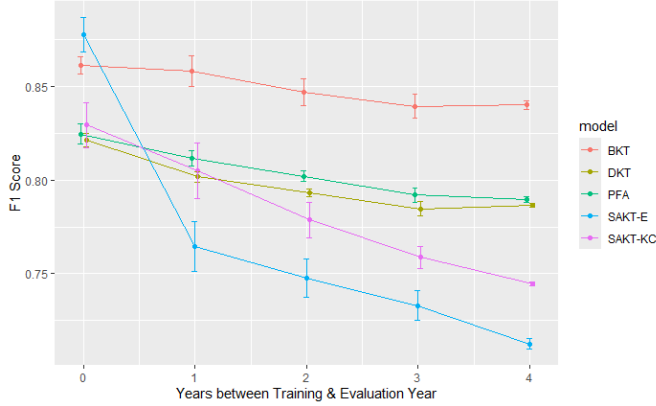


Figure 6: Mean F1 score measurements vs. training data age. Error bars represent a 95% CI for given training data age.

of model degradation, with the caveat that they did not perform as well as other any other models to begin with. Finally, SAKT-E and SAKT-KC both have significant predictive power the first 1-2 years after they are trained, however appear to be degrading with no signs of stopping suggesting they suffer the most from concept drift.

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. [22]. While SAKT-E exhibited the steepest drop-off in model performance over time, particularly in the year immediately following its training, SAKT-KC also experienced statistically significant losses in AUC despite having orders of magnitude fewer tunable parameters. DKT which had more parameters than SAKT-KC but fewer than SAKT-E, and was far more robust than both. BKT and PFA, which had the fewest tunable parameters by orders of magnitude, experienced a noticeable but minimal degradation. Rather, the main factor which contributes to model degradation appears to be the presence of the attention mechanism in both SAKT-KC and SAKT-E. SAKT-E is the top performing model and SAKT-KC is the next-best model by

Model	AUC	Logistic Loss	F1 Score
BKT	-0.374	0.411	-0.428
PFA	-0.716	0.779	-0.721
DKT	-0.917	0.918	-0.861
SAKT-E	-0.883	0.872	-0.810
SAKT-KC	-0.695	0.693	-0.673

Table 4: Estimated Spearman’s ρ for each model fit statistic. All estimated coefficients have $p < 0.001$.

Coefficients	Estimate	Std. Err	p-value
Years Between (YB)	-0.010	2.12e-3	1.63e-5
BKT	0.842	3.87e-3	
PFA	0.729	3.87e-3	
DKT	0.743	3.87e-3	
SAKT-E	0.921	3.87e-3	
SAKT-KC	0.879	3.87e-3	
YB×PFA	-4.95e-3	3.00e-3	0.496
YB×DKT	-9.70e-3	3.00e-3	7.58e-3
YB×SAKT-E	-0.039	3.00e-3	1.01e-33
YB×SAKT-KC	-0.018	3.00e-3	2.94e-8

Table 5: Regression results for AUC (adjusted $R^2 = 0.836$), with BKT as reference category

AUC the when evaluated on the year in which they were trained, however, both exhibit steep decline as the years between evaluation and training increase.

Given that the two SAKT models had significantly higher rates of performance decline than the other three models, it seems that the attention mechanism of SAKT may cause its downfall. One advantage of self-attentive models in other domains is their ability to capture long-range dependencies. This could serve as a detriment since students’ performance is more correlated to their performance on nearby problems rather than how they did on problems two months ago. Further, this would explain, in part, why across years the SAKT models perform significantly worse, as longer range patterns derived from one year lose their explanatory power in future years due to curriculum changes, ordering of skills taught, and other sources of concept shift. Such close-range dependencies would favor BKT and DKT due to their ability to estimate ability on certain skills and how students have performed on recent problems.

5.1 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. Alternatively, a more thorough investigation of the learned attention weights of the different SAKT models could yield more insight, either identifying what parts of the learning process change over time or verifying that the model is more sensitive to noisy sequences. We limited our analysis to basic implementations of our four KT models. As

Coefficients	Estimate	Std. Err	p-value
Years Between (YB)	0.016	3.51e-3	5.32e-5
BKT	0.433	6.41e-3	
PFA	0.562	6.41e-3	
DKT	0.545	6.41e-3	
SAKT-E	0.312	6.41e-3	
SAKT-KC	0.378	6.41e-3	
YB×PFA	5.70e-3	4.96e-3	0.597
YB×DKT	1.62e-3	4.96e-3	0.745
YB×SAKT-E	0.070	4.96e-3	4.75e-39
YB×SAKT-KC	0.025	4.96e-3	5.21e-6

Table 6: Regression results for Log Loss (adjusted $R^2 = 0.737$), with BKT as reference category

Coefficients	Estimate	Std. Err	p-value
Years Between (YB)	-6.55e-3	1.82e-3	2.33e-3
BKT	0.862	3.32e-3	
PFA	0.823	4.69e-3	
DKT	0.817	4.69e-3	
SAKT-E	0.850	4.69e-3	
SAKT-KC	0.828	4.69e-3	
YB×PFA	-3.31e-3	2.57e-3	0.597
YB×DKT	-3.96e-3	2.57e-3	0.496
YB×SAKT-E	-0.038	2.57e-3	9.50e-42
YB×SAKT-KC	-0.016	2.57e-3	6.13e-9

Table 7: Regression results for F1 score (adjusted $R^2 = 0.632$), with BKT as reference category

discussed in Shen et al. [34], 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.

6. CONCLUSION

Expanding upon previous findings, this study indicates that many popular families of Knowledge Tracing models can lose predictive power over time. BKT and PFA exhibit the slowest rates of decline, while more recently proposed attention-based models decline faster. These findings indicate that the underlying process of student learning (as monitored through student interaction logs) may not be as stable as previously theorized.

These results have multiple implications for researchers and practitioners in the EDM field. There is a long-standing discussion surrounding the pros and cons of deep learning models in an educational context. The choice between “simpler”/more explainable models and more complex but less explainable models involves considerations of data availability, the desire for explainable models, and ease of model fitting, among other concerns. Based on our findings, we posit that model longevity should be another consideration when deciding how to model student learning. If a model is needed only for the short-medium term, a deeper model may well be more appropriate. In situations where a model may need to be valid over a longer duration, our results indicate a simpler model may be a better choice. Deeper models may indeed be more performant when applied to a data-

abundant environment, and when refit every two-to-three, but in more data-sparse contexts, simpler models may be more advisable. We encourage other researchers interested in proposing new KT models to explore their models’ fitness over longer time frames rather than relying on a single benchmark dataset, and in the interest of pursuing this goal, we are releasing the samples described in section 3.1 with this paper.

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

Redacted for blinding purposes.

7. ADDITIONAL AUTHORS

8. REFERENCES

- [1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- [2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization, 2016.
- [3] R. S. J. D. Baker, A. T. Corbett, and V. Aleven. More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing. In 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, editors, *Intelligent Tutoring Systems*, volume 5091, pages 406–415. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. Series Title: Lecture Notes in Computer Science.
- [4] R. S. J. d. Baker, A. Mitrović, and M. Mathews. Detecting Gaming the System in Constraint-Based Tutors. In P. De Bra, A. Kobsa, and D. Chin, editors, *User Modeling, Adaptation, and Personalization*, pages 267–278, Berlin, Heidelberg, 2010. Springer.
- [5] B. S. Bloom. 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 1968. Publisher: Regional Education Laboratory for the Carolinas and Virginia, Mutual Plaza (Chapel Hill and Duke Sts.), Durham, N.C. 27701.
- [6] A. F. Botelho, R. S. Baker, and N. T. Heffernan. Improving Sensor-Free Affect Detection Using Deep Learning. In E. André, R. Baker, X. Hu, M. M. T. Rodrigo, and B. du Boulay, editors, *Artificial*

- Intelligence in Education*, pages 40–51, Cham, 2017. Springer International Publishing.
- [7] R. A. Calvo and S. D’Mello. Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications. *IEEE Transactions on Affective Computing*, 1(1):18–37, Jan. 2010. Conference Name: IEEE Transactions on Affective Computing.
 - [8] H. Cen, K. Koedinger, and B. Junker. Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement. In M. Ikeda, K. D. Ashley, and T.-W. Chan, editors, *Intelligent Tutoring Systems*, pages 164–175, Berlin, Heidelberg, 2006. Springer.
 - [9] W. Chu and P. I. P. Jr. The Predictiveness of PFA is Improved by Incorporating the Learner’s Correct Response Time Fluctuation. July 2023. ISBN: 9781733673648 Pages: 244–250 Publication Title: Proceedings of the 16th International Conference on Educational Data Mining Publisher: International Educational Data Mining Society.
 - [10] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, Dec. 1994.
 - [11] O. B. Deho, L. Liu, J. Li, J. Liu, C. Zhan, and S. Joksimovic. 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, 2024. Conference Name: IEEE Transactions on Learning Technologies.
 - [12] R. Donnelly and H. A. Patrinos. Learning loss during Covid-19: An early systematic review. *PROSPECTS*, 51(4):601–609, Oct. 2022.
 - [13] T. Gervet, K. Koedinger, J. Schneider, and T. Mitchell. When is Deep Learning the Best Approach to Knowledge Tracing? *Journal of Educational Data Mining*, 12(3):31–54, Oct. 2020. Number: 3.
 - [14] P. Hallinger and R. H. Heck. Exploring the journey of school improvement: classifying and analyzing patterns of change in school improvement processes and learning outcomes. *School Effectiveness and School Improvement*, 22(1):1–27, 2011.
 - [15] 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*, 24(4):470–497, Dec. 2014.
 - [16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.
 - [17] C. Huyen. *Designing Machine Learning Systems*, chapter Data Distribution Shifts and Monitoring, pages 225–262. O’Reilly Media, 2022.
 - [18] M. Khajah, R. V. Lindsey, and M. C. Mozer. How deep is knowledge tracing? In *arXiv.org*, Mar. 2016.
 - [19] R. Klinkenberg and T. Joachims. Detecting concept drift with support vector machines. 2000.
 - [20] Z. J. Kovačić. Early prediction of student success: Mining students’ enrolment data. In *Proceedings of Informing Science & IT Education Conference (InSITE)*, 2010.
 - [21] M. P. Lee, E. Croteau, A. Gurung, A. F. Botelho, and N. T. Heffernan. Knowledge Tracing over Time: A Longitudinal Analysis. International Educational Data Mining Society, 2023. ERIC Number: ED630851.
 - [22] N. Levin, R. Baker, N. Nasir, F. Stephen, and S. Hutt. Evaluating Gaming Detector Model Robustness Over Time. *Proceedings of the 15th International Conference on Educational Data Mining, International Educational Data Mining Society*, Jan. 2022.
 - [23] S. Madireddy, P. Balaprakash, P. Carns, R. Latham, G. K. Lockwood, R. Ross, S. Snyder, and S. M. Wild. Adaptive Learning for Concept Drift in Application Performance Modeling. In *Proceedings of the 48th International Conference on Parallel Processing, ICPP ’19*, pages 1–11, New York, NY, USA, Aug. 2019. Association for Computing Machinery.
 - [24] M. Nicolaidou and M. Ainscow. Understanding Failing Schools: Perspectives from the inside. *School Effectiveness and School Improvement*, 16(3):229–248, Sept. 2005. Publisher: Routledge eprint: <https://doi.org/10.1080/09243450500113647>.
 - [25] S. Pandey and G. Karypis. A Self-Attentive model for Knowledge Tracing, July 2019. arXiv:1907.06837 [cs, stat].
 - [26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.
 - [27] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance Factors Analysis – A New Alternative to Knowledge Tracing. Technical report, 2009. Publication Title: Online Submission ERIC Number: ED506305.
 - [28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
 - [29] 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*, volume 28. Curran Associates, Inc., 2015.
 - [30] T. S. Popkewitz. Educational Standards: Mapping Who We Are and Are to Become. *Journal of the Learning Sciences*, 13(2):243–256, Apr. 2004.
 - [31] Y. Qiu, Y. Qi, H. Lu, Z. Pardos, and N. Heffernan. Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing. pages 139–148, Jan. 2011.
 - [32] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett. Cognitive Tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, 14(2):249–255, Apr. 2007.
 - [33] A. Shemshack and J. M. Spector. A systematic literature review of personalized learning terms. *Smart*

Learning Environments, 7(1):33, Oct. 2020.

- [34] S. Shen, Q. Liu, Z. Huang, Y. Zheng, M. Yin, M. Wang, and E. Chen. A Survey of Knowledge Tracing: Models, Variants, and Applications. *IEEE Transactions on Learning Technologies*, 17:1898–1919, 2024. Conference Name: IEEE Transactions on Learning Technologies.
- [35] K. Vanacore, A. Gurung, A. Sales, and N. T. Heffernan. The effect of assistance on gamers: Assessing the impact of on-demand hints & feedback availability on learning for students who game the system. In *Proceedings of the 14th Learning Analytics and Knowledge Conference*, pages 462–472, 2024.
- [36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need, June 2017. arXiv:1706.03762.
- [37] J.-J. Vie and H. Kashima. Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):750–757, July 2019. Number: 01.
- [38] X. Wang, Z. Wang, W. Shao, C. Jia, and X. Li. Explaining Concept Drift of Deep Learning Models. In J. Vaidya, X. Zhang, and J. Li, editors, *Cyberspace Safety and Security*, pages 524–534, Cham, 2019. Springer International Publishing.
- [39] Z. Wang, X. Feng, J. Tang, G. Y. Huang, and Z. Liu. Deep Knowledge Tracing with Side Information. In S. Isotani, E. Millán, A. Ogan, P. Hastings, B. McLaren, and R. Luckin, editors, *Artificial Intelligence in Education*, pages 303–308, Cham, 2019. Springer International Publishing.
- [40] K. Williamson and R. F. Kizilcec. Effects of Algorithmic Transparency in Bayesian Knowledge Tracing on Trust and Perceived Accuracy. Technical report, International Educational Data Mining Society, 2021. ERIC Number: ED615541.
- [41] C.-K. Yeung. Deep-IRT: Make Deep Learning Based Knowledge Tracing Explainable Using Item Response Theory, Apr. 2019. arXiv:1904.11738.
- [42] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized Bayesian Knowledge Tracing Models. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Artificial Intelligence in Education*, pages 171–180, Berlin, Heidelberg, 2013. Springer.
- [43] 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*, L@S ’17, pages 169–172, New York, NY, USA, Apr. 2017. Association for Computing Machinery.

APPENDIX

A. MODEL EXPLANATIONS

A.1 BKT

$$P(L_1) = P(L_0)$$

$$P(L_t|C_t = 1) = \frac{P(L_t) \cdot (1 - P(S))}{P(L_t) \cdot (1 - P(S)) + (1 - P(L_t)) \cdot P(G)}$$

$$P(L_t|C_t = 0) = \frac{P(L_t) \cdot P(S)}{P(L_t) \cdot P(S) + (1 - P(L_t)) \cdot (1 - P(G))}$$

$$P(L_{t+1}) = P(L_t|C_t) \cdot (1 - P(F)) + (1 - P(L_t|C_t)) \cdot P(T)$$

$$P(C_{t+1}) = P(L_{t+1}) \cdot (1 - P(S)) + (1 - P(L_{t+1})) \cdot P(G)$$

$P(L_t)$ is the probability that the student has mastered the relevant KC at time step t , $P(C_t)$ is the probability of said student getting the next exercise of that KC correct. $P(L_0)$, $P(G)$, $P(S)$, $P(T)$, and $P(F)$ are the probabilities for prior knowledge, guess, slip, transfer (learning), and forgetting, respectively, and are learned via expectation maximization [10].

A.2 PFA

$$m(i, j \in KCs, s, f) = \sum_{j \in KCs} \beta_j + \gamma_j s_{i,j} + \rho_j f_{i,j}$$

$$P(m) = \frac{1}{1 + e^{-m}} = \sigma(m)$$

In this model, i represents a student, while j represents a KC. $s_{i,j}$ and $f_{i,j}$ represent collective successes and failures of student i on KC j [27].

A.3 DKT

Our implementation of DKT uses the standard LSTM [16] fed into a dense output layer. An LSTM computes the following function:

$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi})$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf})$$

$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg})$$

$$o_t = \sigma((W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}))$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

$$h_t = o_t \odot \tanh(c_t)$$

where i_t , f_t , g_t , o_t , c_t , h_t are the input, forget, output, cell, and hidden state at time t , x_t is the model input at time t , σ is the sigmoid function and $\odot$ is the element-wise product. All instances of W and b are trainable parameters.

A.4 SAKT

Our implementation of SAKT uses architecture initially described in Pandey et al’s SAKT paper [25]. The architecture using exercises as features is fittingly called SAKT-E. An added change from the original paper is the addition of KCs as a potential feature for embedding, for which the architecture is appropriately titled SAKT-KC.

A.4.1 Embedding Layers

$$E_x = \{e_t \mid e_t = \hat{E}[E_t], E_t \in \{0, 1\}\}, \quad \hat{E} \in \mathbb{R}^{B \times T \times d}$$

$$R_x = \{r_t \mid r_t = \hat{R}[R_t], R_t \in \{0, 1\}\}, \quad \hat{R} \in \mathbb{R}^{B \times 2 \times d}$$

$$P_x = \{p_t \mid p_t = E_g[t], t \in [1, T]\}, \quad E_g \in \mathbb{R}^{B \times T \times d}$$

$$Z = E_x + R_x + P_x$$

Notations	Descriptions
B	Batch size
T	Sequence length
d	Embedding dimension
K	Total number of exercise/KC IDs
X	Past learner responses, $X \in \mathbb{R}^{B \times T \times 2}$
R	Past binary responses $R \in \{0, 1\}$
y	Ground truth labels ($y \in \{0, 1\}$)
C	Binary ground truth labels ($C \in \{0, 1\}$)
L	Number of unique exercises
E_x	Trainable exercise/KC ID embedding matrix
P_x	Trainable position embedding matrix
R_x	Trainable Response embedding matrix
W	Trainable Weight Matrix

Table 8: Notations for SAKT Architecture

A.4.2 Self-Attention Mechanism

$$Q = ZW^Q, \quad K = ZW^K, \quad V = ZW^V$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V$$

$$\text{Head}_h = \text{Attention}(QW_h^Q, KW_h^K, VW_h^V)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{Head}_1, \dots, \text{Head}_H)W^O$$

Here, softmax refers to the softmax function, formally defined as

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

where n is the number of classes.

A.4.3 Residual Connections and LayerNorm

$$O = \text{LayerNorm}(Z + \text{MultiHead}(Q, K, V))$$

$$F(O) = \max(0, OW_1 + b_1)W_2 + b_2$$

$$O = \text{LayerNorm}(O + F(O))$$

$$\hat{y}_t = \sigma(W_y O_t + b_y)$$

LayerNorm here refers to the function initially defined in Ba, et.al's paper [2] on normalization of neurons, which reduces training time significantly. This formula is given by $y = \frac{x - E[x]}{\sqrt{\text{Var}(x) + \epsilon}}$ where ϵ is a small value added to avoid division by 0. In the calculation of our binary prediction $\hat{y}$, σ represents the sigmoid activation function, or $\frac{1}{1+e^{-x}}$ to squash the output into the range of (0, 1) for evaluation with the loss function.

A.4.4 Loss Function

$$\mathcal{L} = -\frac{1}{K} \sum_{i=1}^K [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Here, we use the standard binary cross entropy loss, with K representing the number of observations.