Expert Features for a Student Support Recommendation Contextual Bandit Algorithm

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

Abubakir Siedahmed
Worcester Polytechnic Institute
Worcester, Massachusetts, USA
asiedahmed@wpi.edu

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

ABSTRACT

Contextual multi-armed bandits have previously been used to personalize student support messages given to learners by supplying a model with relevant context about the user, problem, and available student supports. In this work, we propose using careful feature selection with relevant domain knowledge to improve the quality of student support recommendations. By providing Bayesian Knowledge Tracing mastery estimates to a contextual multi-armed bandit as user-level context in a simulated environment, we demonstrate that using domain knowledge to engineer contextual features results in higher average cumulative reward, and significant improvement over randomly selecting student supports. The data used to simulate sequential recommendations are available at https://osf.io/sfyzv/?view_only=351fb8781d2c4f3bbc9d7486762d563a.

CCS CONCEPTS

• **Applied computing** → **Computer-managed instruction**; • **Computing methodologies** → *Simulation evaluation*; **Sequential decision making**.

KEYWORDS

Reinforcement Learning, Feature Engineering, Multi-Armed Bandit, Personalized Learning

ACM Reference Format:
Morgan P Lee, Abubakir Siedahmed, and Neil T. Heffernan. 2024. Expert Features for a Student Support Recommendation Contextual Bandit Algorithm. In *The 14th Learning Analytics and Knowledge Conference (LAK '24), March 18–22, 2024, Kyoto, Japan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3636555.3636909

1 INTRODUCTION

The drive towards personalization has been a foundational tenet in advancing the research and methodologies within the Learning Analytics community [28]. While personalized learning research is a broad sub-field, one core aspect of personalized learning research involves the tailoring of instruction and remediation to individual students' needs [23]. More specifically, personalized learning involves a qualitative interaction between two or more groups of students who benefit from different kinds of instruction (i.e. group A of students benefits more from teaching method 1 than method 2, while group B of students benefits more from method 2 than method 1). Finding and exploiting these qualitative interactions to benefit students is a desirable goal, and randomized experiments comparing different educational interventions have been incorporated into a number of online learning platforms to attempt to find these qualitative interactions [20–22]. In an ideal setting, these interactions can be exploited to improve student learning as soon as they are found, but due to the nature of A/B testing, students are held in condition for the duration of an experiment, meaning that if one intervention proves to be more effective, students in other conditions are at a disadvantage through no fault of their own.

Reinforcement learning (RL) is a common machine learning approach that is well-suited to many different personalization tasks, including instructional sequencing [7] and feedback generation [1]. It also provides a potential solution to unfair treatment of students during A/B testing through adaptive experimentation [24]. Specifically, the use of multi-armed bandit algorithms (MABs) can both learn these qualitative interactions and make use of them to improve student learning outcomes. This is often done by supplying a MAB with context about the learner and learning environment. Contextual multi-armed bandit algorithms (CMABs) require side information about students, problems, and potential interventions in order to both learn which interventions are better in what contexts, and to identify the appropriate intervention to give in a provided context. The question of what features to supply to a CMAB as context is vital.

In this paper, we explore the effects of different features on the quality of CMAB recommendations in a simulated environment. We propose the use of expert features as context to a student support recommendation CMAB, namely the output of another well-studied technique for evaluating student knowledge: Bayesian Knowledge Tracing (BKT). We then conducted multiple simulations of CMABs using standard average features and BKT knowledge estimates to explore the effectiveness of expert features on recommendation quality.

2 BACKGROUND

2.1 Contextual Multi-Armed Bandit Algorithms

Multi-armed bandit algorithms are RL agents which choose one action out of an available set of actions. A reward function is defined in relation to these actions as a way to incentivize "good" behavior by the agent, which learns over time the relationships between actions and their rewards. A key assumption of MABs is that actions are independent: the sequence of actions has no impact on potential rewards. This makes MABs less computationally complex than other RL decision-making agents.

LAK '24, March 18–22, 2024, Kyoto, Japan

ACM ISBN 979-8-4007-1618-8/24/03.
https://doi.org/10.1145/3636555.3636909

Depending on the nature of the problem, supplying a MAB with side information can allow the agent to make more informed decisions. These Contextual Multi-Armed Bandits have been studied in a number of content recommendation environments [13], and have been utilized to great effect in multiple contexts within intelligent tutoring systems (ITS), including problem sequencing [5], tutoring action recommendation [14], and student support (hint/explanation) recommendation [18].

Supplying MABs with context introduces a host of additional challenges. Different modeling techniques can be used to predict future rewards, ranging from linear regression to deep learning [26]. The trade-offs between these modeling techniques are much the same for CMABs as for other domains: regressions require less computational power and are more easily explainable, while neural networks can exploit nonlinear relationships that are missed by a one-layer model. CMABs must also balance exploration of the relationship between context and reward with exploiting previously learned information, the well-studied exploration/exploitation trade-off [3].

2.2 Expert Features

While the choice of which CMAB to implement is clearly important, the specific context provided to a bandit algorithm is equally worthy of consideration. Previous implementations of CMABs in ITS have incorporated student-level features about how students interact with the ITS such as prior exercise performance [5], median first response time, and problem completion percentage [18, 19]. If available, student demographic information or problem-specific context may also be encoded as features. However, including unnecessary features has the potential to not only decrease model performance, but to disadvantage students with less-common feature values [12]. Thus, the benefits and risks of adding or changing features of a CMAB algorithm must be carefully considered.

The use of expert engineered features in other machine learning domains has often yielded similar results to deep learning methods when applied to student modeling problems [2, 9]. The idea of utilizing engineered features as context in a CMAB recommender system has yet to be explored based on the available literature, but the primary barrier to using feature engineering for this problem is domain knowledge. What features should be provided to a model to allow for personalization? Empirical studies of student support recommendation systems have found that context related to prior knowledge was the most critical for achieving qualitative interactions between different student supports and groups of students [19]. Features which encode information about prior knowledge, then, are vital context to achieve personalization.

2.2.1 Knowledge Tracing. Previous models have encoded prior knowledge as an average of prior problem correctness [18, 19]. While this is a rough measure of a student's general knowledge, predicting student knowledge of particular knowledge concepts, or Knowledge Tracing (KT), is a well-established problem in educational data mining. Techniques for assessing student knowledge states have involved Hidden Markov Models [6], logistic regression [4, 16, 25], recurrent neural networks [17], dynamic key-value memory networks [27], and attention-based transformer models [15]. Explainability is a key concern in student support recommender systems. Given its interpretability and cognition-based model of knowledge acquisition and retention, Bayesian Knowledge Tracing (BKT), which models student knowledge of skills as a latent (hidden) variable in a Markov chain, could potentially serve as a better feature of student knowledge than an average. Moreover, since BKT makes predictions on a per-skill basis, student knowledge of different skills is modeled independently.

2.3 ASSISTments

The data used in this work came from ASSISTments. ASSISTments is a free-to-use online learning platform which allows teachers to assign problem sets from open-source curricula to their students. Students must correctly answer a given exercise before continuing on, and students are able to request help in the form of written hints and explanations.

2.3.1 Problem-Level Support. Most problems within ASSISTments are associated with between two to four problem-level supports in the form of either sets of hints or explanations. Hint sets contain multiple smaller bits of tutoring that the student can request in sequence, forfeiting a portion of their final score on a question with each requested hint. Explanations are full, complete solutions to the given problem and contain the correct answer. Requesting an explanation forfeits all credit for the exercise.

2.3.2 Student Support Delivery Service. At the beginning of a problemset, the Student Support Delivery Service determines student supports to be made available to the student based on a variety of factors. Relevant to this study, sometimes a student will be given a random selection of the available student supports for a given problem set. Students can receive a set of hints, an explanation, or simply the correct answer for each problem within the problem set, but only one of these options will be made available to the student on each problem. Hint sets usually contain several parts which attempt to break down the problem into smaller parts, while explanations contain the answer to the problem, often in the form of a worked example.

3 METHODS

We propose the use of expert features as contextual input to a CMAB recommender agent. Specifically, the use of Bayesian Knowledge Tracing with a forgetting parameter to estimate student knowledge of the relevant skill. This knowledge estimation is given as a user-level feature to a CMAB agent, alongside problem and tutoring strategy-level features. To examine the impact of BKT state predictions as contextual input, a simulation study based on random student support recommendations was conducted.

3.1 Data Collection

Data for this study was collected from the Student Support Delivery Service within the ASSISTments platform between May 2021 and September 2023. The primary data used to simulate a recommendation agent came from these randomly assigned tutor strategies merged with relevant problem, user, and skill information. These random instances were then filtered by whether or not the student requested help on the associated problem, and whether or not the student had completed another question as part of the same

assignment. The final simulation dataset yielded 330,071 rows representing 40,711 students interacting with 5,767 different problems tagged with 190 distinct skills. Each row represents a student viewing a student support on a single problem. A full description of the information available in the simulation dataset can be found in appendix A. Since every tutor strategy in each row of the final simulation dataset was given at random, we can use this data as a representation of the population distribution of the ASSISTments userbase.

3.2 Model Selection

Multiple CMAB algorithms were considered for implementation in this study. In order to make the impact of feature selection more apparent, only one CMAB was simulated, with the differences between simulated runs being limited to the features considered by a variant of the same underlying model. In the context of student support recommendation, model explainability was prioritized, limiting our search space to regression-based approaches to CMABs. Due to its efficacy in previous simulations and its prior empirical study in Prihar, Sales, & Heffernan [19], Dynamic Linear Epsilon Greedy was chosen.

Since one of the simulations required state predictions from a BKT model as context, a BKT model with an enabled forgetting parameter was trained on available student performance data collected during the 2020-2021 school year in the ASSISTments platform. While forgetting parameters have often been manually disabled in BKT models since their inception, explicitly modeling forgetting behavior often grants a sizable performance boost [10]. Performance data from the 20-21 year was chosen for two reasons. First, since the simulation data contains data from the 2021-2022 school year onwards, the 20-21 school year performance data is disjoint from the data used in simulation. Second, prior work has demonstrated BKT models to retain sufficient generalizability in the short-medium term [11], meaning a BKT model trained on the 20-21 school year would likely be transferable to future years. The final BKT model had an accuracy of 0.733 and an AUC of 0.757 when evaluated on its training data.

After fitting the BKT model, all problem logs containing a (student, skill) pair found in the simulation data were collected. BKT state predictions were computed at every relevant timestep, and then merged with the simulation data to provide the BKT state estimation on the associated skill for every row of the simulation data.

3.3 Simulation Design

The following protocol was used to simulate each CMAB making a series of sequential recommendations:

(1) Initialize the CMAB.
(2) Select a single instance of a student recieving and viewing support from the SSDS.
(3) Provide relevant user, problem, and support features to the CMAB.
(4) Have the CMAB recommend a support for the hypothetical student to receive.
(5) If the support given by the CMAB matches the support given by the SSDS, update the CMAB using the next problem correctness as the reward. Otherwise, repeat from step 2.
(6) Continue repeating steps 2-4 until the desired number of recommendations have been made by the CMAB.

One run of a simulation consists of 1,000,000 sequential recommendations with updates. Three different models were simulated: one DLEG bandit using prior percent correct as the user feature, one DLEG bandit using BKT state predictions as the user feature, and one random selection model. Each model was simulated five times to evaluate the significance of any differences in the cumulative reward distributions of each model.

Table 1: A comparative analysis of the three models (Random, DLEG, and DLEG & BKT) based on average, maximum, and minimum

	Minimum	Average	Maximum
DLEG + BKT	339,285	340,867	343,132
DLEG	335,986	338,031	340,704
Random	336,016	336,461	336,813

4 RESULTS

Figure 1 illustrates a comparative analysis of three models – random, DLEG, and DELG & BKT – with 95% confidence intervals. These intervals depict the variability in the mean of each model, highlighting the potential range of performance outcomes. Wider intervals, as observed in the DLEG model, signify greater uncertainty, while narrower intervals, as seen in the random model, suggest less uncertainty and more reliable results.

The 95% confidence interval for the random model spans from 336,050.17 to 336,871.03, indicating a high level of precision. In contrast, the DLEG model exhibits a broader interval, ranging from 335,767.87 to 340,293.73, suggesting increased uncertainty. Similarly, the 95% confidence interval for the DLEG & BKT model extends from 338,986.75 to 342,748.05, indicating a considerable degree of uncertainty.

Notably, a slight overlap in the confidence intervals exists for the DLEG & BKT model and the DLEG model. This overlap, though small, suggests enough significant difference to further investigate the models. Thus, we aggregated the reward to find the average, maximum, and minimum for the three models.

Table 1 shows the average, maximum, and the minimum rewards for the models. For instance, the DLEG & BKT model demonstrated an average reward of 340,867, a maximum of 343,132, and a minimum of 339,285 across the five simulation runs. In context, this means that out of the 1 million recommendations that were given as part of a simulation, the student got their next exercise correct 343,132 times in one of the simulations running the DLEG & BKT model.

To further investigate the differences between the simulation types, we conducted an ANOVA to try and detect a difference between the mean cumulative rewards. The results of this analysis are in table 2. The test concludes that at least one of the simulation types has a mean that differs from the other two. To investigate

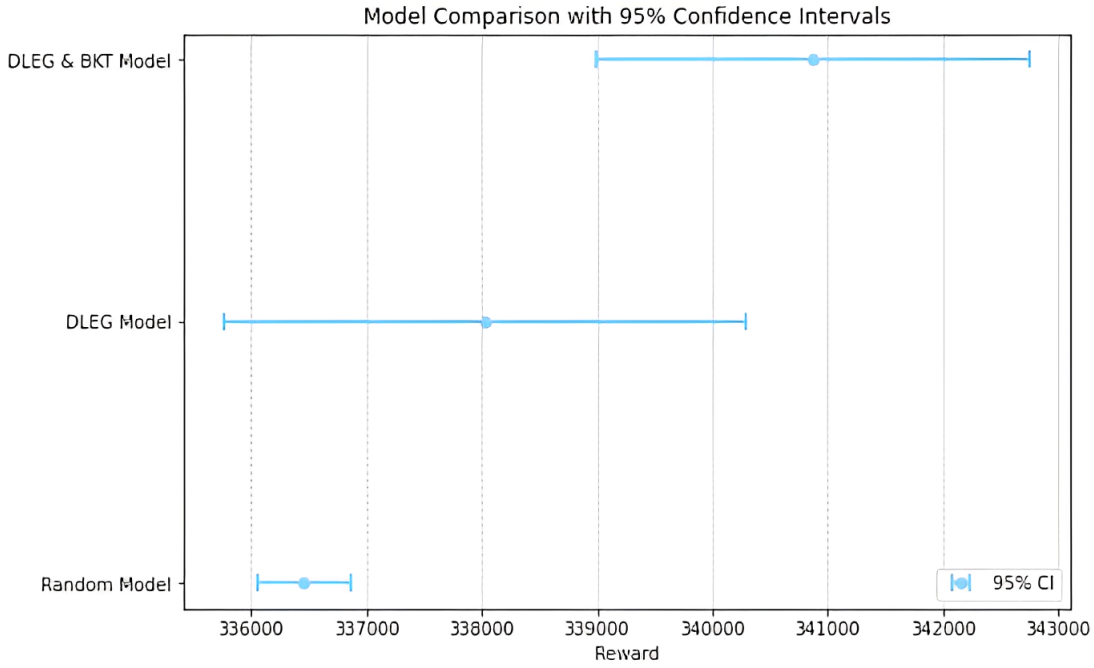

Figure 1: An error bar plot comparing the mean reward and 95% confidence intervals of the three models (Random, DLEG, and DLEG & BKT) to provide overview of their performance.

this further, we conducted a Tukey multiple comparisons test to examine the differences between each simulation type. The results of the Tukey test can be found in table 3. We found that the DLEG + BKT model was significantly different than both the random model and the standard DLEG model, while the difference between standard DLEG and random was not significant. A plot of these mean differences and their respective confidence intervals can be found in figure 2.

5 DISCUSSION

In this study, we examined the influence of various features on the quality of recommendations generated by CMAB algorithms within a simulated environment. Our findings reveal the advantages of using DLEG & BKT together: we found a statistically significant difference between the standard DLEG model and DLEG with BKT as a user feature. Though this difference is statistically significant, it is a marginal one: we estimate that DLEG+BKT only outperforms DLEG by around 3000 cumulative reward. Stated another way, students in a DLEG+BKT simulation got their next problem correct after looking at a student support 3000 more times than students in a standard DLEG simulation. The differences between DLEG+BKT and random selection were larger, but still only amount to a difference of around 4500 cumulative reward.

However, it is important to acknowledge the limitations of our work, which offer avenues for future research. First, our features were static, and not computed at every timestep. Future work can dynamically compute features at each timestep, which could enhance the adaptability of the model. Given that users differ at every timestep, tailoring features to individual users could significantly contribute to the model's efficacy. Implementing a dynamic feature computation would be beneficial because we'd be able to assess each user's features at each timestep. Additionally, our simulation relied on randomly sampled data, and students' support assignments were random.

Furthermore, integrating the Item Response Theory (IRT) model as a measure of difficulty holds promise [8]. At the moment, the best measure of difficulty we have is relying on the average correctness/incorrectness of a problem. Incorporating IRT could help elicit more accurate and reliable insights, and improve the overall performance of the recommendation system.

Finally, while our simulation's results show promise, and our simulations were conducted by resampling actual student data, our results were in a simulation. It remains to be seen if the gains seen in simulation could transfer to actual students, and an empirical study implementing DLEG+BKT would be necessary to fully assess the effectiveness of this method.

As discussed in the Background section, the exploration of using engineered features in the context of CMAB recommendation system is an underexplored area. However, our work has shown promising results by incorporating expert features using the BKT model with CMAB, instead of a traditional average-based approach. These preliminary results validate the advantages of using domain-specific features, and pave the way for further exploration. We anticipate the results could enhance the reinforcement learning system and contribute to a more personalized and effective recommendation system.

Table 2: ANOVA table comparing different simulation types

	Degrees of Freedom	Sum of Squares	Mean Square	F Value	p-value
Simulation Type	2	49886190	24943095	13.07	0.00097
Residuals	12	22899305	1908275		

Table 3: Results from a Tukey multiple comparisons test

Simulation Type	Difference in Means	Lower Bound	Upper Bound	p-value (adjusted)
(DLEG+BKT) - DLEG	2836.6	505.7509	5167.4491	0.0178
Random - DLEG	-1570.2	-3901.0491	760.6491	0.2120
Random - (DLEG+BKT)	-4406.8	-6737.6491	-2075.9509	0.0008

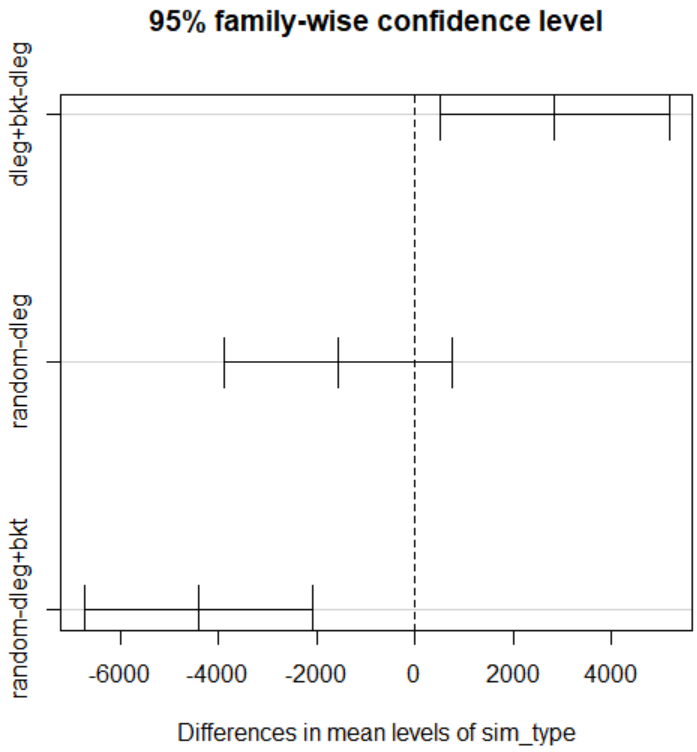

Figure 2: Confidence intervals for the differences between simulation type means.

6 CONCLUSION

In this work, we proposed the use of BKT mastery estimations as a user-level feature in a CMAB to recommend student support messages to learners. Through multiple simulations, we demonstrated that using BKT predictions as a feature significantly improves model performance over random selection and averages as user-level context, though these improvements are marginal. This raises interesting questions about the possibility of using other expert features as improved context, such as IRT covariates as a possible problem-level feature. Future work examining expert features for CMAB algorithms can expand on this concept by running adaptive experiments to empirically validate the phenomena seen in simulation.

ACKNOWLEDGMENTS

Neil and Cristina Heffernan thank their many past and current funders including NSF (2118725, 2118904, 1950683, 1917808, 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, & 1535428), IES (R305N210049, R305D210031, R305A170137, R305A170243,

R305A180401, R305A120125 & R305R220012), GAANN (P200A120238, P200A180088, P200A150306 & P200A150306), EIR (U411B190024 S411B210024, & S411B220024), ONR (N00014-18-1-2768), NHI (via a SBIR R44GM146483), Schmidt Futures, BMGF, CZI, Arnold, Hewlett and a $180,000 anonymous donation. None of the opinions expressed here are those of the funders. We would also like to thank Dr. Adam Sales and Dr. Ethan Prihar for their advice on the statistical analyses conducted in this paper.

REFERENCES

[1] Tiffany Barnes and John Stamper. 2008. Toward automatic hint generation for logic proof tutoring using historical student data. In *International conference on intelligent tutoring systems*. Springer, 373–382.
[2] Anthony F Botelho, Ryan S Baker, and Neil T Heffernan. 2019. Machine-learned or expert-engineered features? Exploring feature engineering methods in detectors of student behavior and affect. In *The twelfth international conference on educational data mining*.
[3] Michael Castronovo, Francis Maes, Raphael Fonteneau, and Damien Ernst. 2013. Learning exploration/exploitation strategies for single trajectory reinforcement learning. In *European Workshop on Reinforcement Learning*. PMLR, 1–10.
[4] Hao Cen, Kenneth Koedinger, and Brian Junker. 2006. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International conference on intelligent tutoring systems*. Springer, 164–175.
[5] Benjamin Clement, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes. 2013. Multi-armed bandits for intelligent tutoring systems. *arXiv preprint arXiv:1310.3174* (2013).
[6] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4 (1994), 253–278.
[7] Shayan Doroudi, Vincent Aleven, and Emma Brunskill. 2019. Where's the reward? a review of reinforcement learning for instructional sequencing. *International Journal of Artificial Intelligence in Education* 29 (2019), 568–620.
[8] Susan E Embretson and Steven P Reise. 2013. *Item response theory*. Psychology Press.
[9] Yang Jiang, Nigel Bosch, Ryan S Baker, Luc Paquette, Jaclyn Ocumpaugh, Juliana Ma Alexandra L Andres, Allison L Moore, and Gautam Biswas. 2018. Expert feature-engineering vs. deep neural networks: which is better for sensor-free affect detection?. In *Artificial Intelligence in Education: 19th International Conference, AIED 2018, London, UK, June 27–30, 2018, Proceedings, Part I 19*. Springer, 198–211.
[10] Mohammad Khajah, Robert V Lindsey, and Michael C Mozer. 2016. How deep is knowledge tracing? *arXiv preprint arXiv:1604.02416* (2016).
[11] MP Lee, E Croteau, A Gurung, A Botelho, and N Heffernan. 2023. Knowledge Tracing Over Time: A Longitudinal Analysis.. In *The Proceedings of the 16th International Conference on Educational Data Mining*.
[12] ZhaoBin Li, Luna Yee, Nathaniel Sauerberg, Irene Sakson, Joseph Jay Williams, and Anna N Rafferty. 2023. Getting too personal (ized): The importance of feature choice in online adaptive algorithms. *arXiv preprint arXiv:2309.02856* (2023).
[13] Tyler Lu, Dávid Pál, and Martin Pál. 2010. Contextual multi-armed bandits. In *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 485–492.
[14] Indu Manickam, Andrew S Lan, and Richard G Baraniuk. 2017. Contextual multi-armed bandit algorithms for personalized learning action selection. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6344–6348.
[15] Shalini Pandey and Jaideep Srivastava. 2020. RKT: relation-aware self-attention for knowledge tracing. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1205–1214.
[16] Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. 2009. Performance Factors Analysis–A New Alternative to Knowledge Tracing. *Online Submission* (2009).
[17] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. *Advances in neural information processing systems* 28 (2015).
[18] Ethan Prihar, Aaron Haim, Adam Sales, and Neil Heffernan. 2022. Automatic Interpretable Personalized Learning. In *Proceedings of the Ninth ACM Conference on Learning@ Scale*. 1–11.
[19] Ethan Prihar, Adam Sales, and Neil Heffernan. 2023. A Bandit You Can Trust. In *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*. 106–115.
[20] Ethan Prithar, Manaal Syed, Korinn Ostrow, Stacy Shaw, Adam Sales, and Neil Heffernan. 2022. Exploring Common Trends in Online Educational Experiments. In *Proceedings of the 15th International Conference on Educational Data Mining*.
[21] Mohi Reza, Juho Kim, Ananya Bhattacharjee, Anna N Rafferty, and Joseph Jay Williams. 2021. The MOOClet framework: unifying experimentation, dynamic improvement, and personalization in online courses. In *Proceedings of the Eighth ACM Conference on Learning@ Scale*. 15–26.
[22] Alexander O Savi, Joseph Jay Williams, Gunter KJ Maris, and Han van der Maas. 2017. The role of A/B tests in the study of large-scale online learning. (2017).
[23] Atikah Shemshack and Jonathan Michael Spector. 2020. A systematic literature review of personalized learning terms. *Smart Learning Environments* 7, 1 (2020), 1–20.
[24] Adish Singla, Anna N Rafferty, Goran Radanovic, and Neil T Heffernan. 2021. Reinforcement learning for education: Opportunities and challenges. *arXiv preprint arXiv:2107.08828* (2021).
[25] Jill-Jênn Vie and Hisashi Kashima. 2019. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 750–757.
[26] Pan Xu, Zheng Wen, Handong Zhao, and Quanquan Gu. 2020. Neural contextual bandits with deep representation and shallow exploration. *arXiv preprint arXiv:2012.01780* (2020).
[27] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*. 765–774.
[28] Ling Zhang, James D Basham, and Sohyun Yang. 2020. Understanding the implementation of personalized learning: A research synthesis. *Educational Research Review* 31 (2020), 100339.

A FEATURE DESCRIPTION

A.1 Simulation Dataset

- **user_xid**: id associated with a particular student user
- **assignment_log_id**: id associated with one student attempting one assignment
- **problem_id**: id associated with a particular problem
- **selected_strategy_id**: id for the tutor strategy assigned by the SSDS
- **alt_strategy_ids**: ids for other tutor strategies that could have been assigned by the SSDS for the given problem
- **problem_log_id**: id associated with one student attempting one problem
- **hint_count**: the number of hints accessed by the student on the given problem
- **bottom_hint**: true if the student was shown the problem's answer by a support, false otherwise
- **skill_id**: id of the skill which tags the current problem
- **start_time**: timestamp of when the student was first shown the current problem
- **discrete_score**: 1 if the student answered the problem correctly on the first attempt, 0 otherwise
- **correct_predictions**: probability that the student will answer their next exercise correctly BEFORE the current exercise was completed, as calculated by a BKT model
- **state_predictions**: probability that the student has the current problem's skill mastered BEFORE the current exercise was completed, as calculated by a BKT model
- **assignment_complete**: true if the student completed every problem in the assignment, false otherwise
- **next_problem_correctness**: discrete_score for the next problem of the student's current assignment

A.2 User Features

- **user_xid**: id associated with a particular student user
- **n**: number of problems completed by the given student
- **correct**: average discrete_score across all problems completed by the given user

A.3 Problem Features

- **problem_id**: id associated with a particular problem
- **n**: number of times the given problem has been completed by students
- **correct**: average discrete_score across all users who have completed the given problem
- **subject_nbt**: 1 if the problem is about numbers and operations in base 10, 0 otherwise
- **subject_ee**: 1 if the problem is about equivalent expressions, 0 otherwise
- **subject_g**: 1 if the problem is about geometry, 0 otherwise
- **subject_ns**: 1 if the problem is about the number system, 0 otherwise
- **subject_oa**: 1 if the problem is about operations and algebraic thinking, 0 otherwise
- **subject_nf**: 1 if the problem is about numbers and operations with fractions, 0 otherwise
- **subject_sp**: 1 if the problem is about statistics and probability, 0 otherwise
- **subject_rp**: 1 if the problem is about ratios and proportional relationships, 0 otherwise
- **subject_md**: 1 if the problem is about measurement and data, 0 otherwise

A.4 Tutor Strategy Features

- **tutor_strategy_id**: id associated with a particular tutor strategy
- **answer_given**: 1 if the tutor strategy contains the problem's answer, 0 otherwise
- **message_count**: normalized representation of the number of messages in a chain of hints
- **ln_character_length**: natural logarithm of the length of the tutor strategy in characters
- **contains_equation**: 1 if the tutor strategy contains an equation, 0 otherwise
- **contains_image**: 1 if the tutor strategy contains an image, 0 otherwise
- **contains_video**: 1 if the tutor strategy contains a video, 0 otherwise