How models for knowledge tracing depend on data

Tikhon Parshikov, Mehrdad Kiani Oshtorjani
Machine learning for behavioural data, CS-421, EPFL, Switzerland

***Abstract*—In this paper, the knowledge tracing on the users of the Lernnavi have been studied. The given raw data was pre-processed and four knowledge tracing methods such as Bayesian Knowledge Tracing (*BKT*), Additive Factor Model (*AFM*), Performance Factor Analysis (*PFA*), and Deep learning knowledge tracing (*DKT*) were applied on the data. The performance of each method have been compared based on root mean square error (*RMSE*) and area under the curve (*AUC*). The training time is another important parameter in this study. *PFA* model showed the best performance but its training time was too high, while *BKT* model trained quicker but the performance was worse.**

I. Introduction

Knowledge tracing is a famous part of machine learning area [1], [2], [3]. Lernnavi is a Swiss website for promoting part of the basic subject-related study skills in German and Mathematics. The following topics are covered in German: "spelling", "punctuation", "parts of speech", "sentence parts", "sentence structure", "stylistics", "text reception", and "text production" [4]. On the website, in the subject of mathematics, at first it focus on the topics of the first two years of the short-term grammar school, namely the following sub-areas: "numbers and sets of numbers", "Term and term transformations", "equations", "functions", "elementary geometry", "resemblance geometry", and "trigonometry". The extension to the themes of the third and fourth years is under development.

The website of Lernnavi provides tasks from sub-areas of the subjects German and Mathematics, which can be processed in three different modes:

"Learn mode": In the "Learn" mode, the subject area can be selected and tasks are assigned to you according to your own abilities. The tasks are selected in such a way that the increase in ability is maximized, so that the students are challenged but not overwhelmed. Various tools are available in the learning mode: you can access a theoretical part and have the opportunity to ask other users questions in the forum. There is help and tips for certain tasks and you receive feedback after each solved task.

"Level check mode": on this mode, the user have the opportunity to check where you stand in relation to your competence in a topic. The user can also select the sub-area in the level check. Here, the tasks are selected in such a way that the greatest possible certainty about the level of knowledge of the students can be gained in the shortest possible time. The learning mode tools are not available in test mode and feedback is only available after the test session is completed.

"Teaching mode": this mode is available to the teachers. They have access to the task pool of Lernnavi and can put together task packages. The teacher decides which tasks are to be processed in which order and the adaptive, personalized assignment of tasks is switched off. Of course, it is always possible to encourage the students to work independently in Lernnavi in class or as homework. By using Lernnavi with under performing students, teachers can be relieved of having to provide additional materials for them.

The data set provided for this work is consist of the information of the users of Lernnavi's for several months and the data is formatted in three main tables:

- *Users*: information about the users of the platform;
- *Events*: all the events done by the users;
- *Transactions*: about questions and answers provided by the users.

But there are also other auxiliary tables that expand the amount of information that can be get from the data set such as: i. difficulty of the problems, ii. sharing or not the results with the teacher, iii. level-check score, iv. accepting the results or not (if the user finally accepted the result of the session), v. the user skipped the question or not, vi. platform is in the foreground or background (the user is doing something else) and many other information.

II. Methodology

The user learning curve is our latent variable in this project which is not directly observable/cannot be measured and it is assumed to affected the outcome of other variables which can be observed (directly measured).

To know if the student is learning or not we used four different methods [5]:

(i) Bayesian Knowledge Tracing (*BKT*) model: is an algorithm used in many intelligent tutoring systems to model each learner's mastery of the knowledge being tutored. It models student knowledge in a hidden Markov model as a latent variable, updated by observing the correctness of each student's interaction in which they apply the skill in question. *BKT* assumes that student knowledge is represented as a set of binary variables, one per skill, where the skill is either mastered by the student or not. Observations in *BKT* are also binary i.e., a student gets a problem/step either right or wrong. Intelligent tutoring systems often use *BKT* for mastery learning and problem sequencing. In its most common implementation, *BKT* has only skill-specific parameters [6]. Before using the *BKT*, the assumption behind the model should be kept in mind such as i. knowledge can be divided into different skills, ii. definition of skills is accurate/detailed enough, iii. each task corresponds to a single skill (original), iv. there is no connection between the skills, v. mastery can be achieved through practice, vi. there is no forgetting: $p_F = 0$ (original).

(ii, iii) Additive Factor (*AFM*) and Performance Factor Analysis (*PFA*) models: The *AFM* and *PFA* models are both based on logistic regression and item response theory (IRT). Specifically, they compute the probability that a student will solve a task correctly based on the number of previous attempts the student had at the corresponding skill (in case of *AFM*) [7], [8], [9], [10] and based on the correct and wrong attempts at the corresponding skill (in case of *PFA*) [11], [12], [13], [14], [15], respectively. The assumptions of *AFM* mode are: i. Students may initially know more or less, ii. Students learn at the same rate, iii. Some skills are more likely to initially be known, iv. Some skills are easier to learn than others, v. Students learn with each practice opportunity, vi. Each item belongs to one or more skills. Moreover, the assumptions behind the *PFA* model are: i. Students may initially know more or less, ii. Students learn at the same rate, iii. Some skills are more likely to initially be known, iv. Some skills are easier to learn than others, v. Students learning rate differs for correct and wrong, vi. practice opportunities, vii. Each item belongs to one or more skills.

(iv) a Deep learning knowledge tracing (*DKT*) model: Knowledge tracing is one of the key research areas for empowering personalized education. It is a task to model students' mastery level of a skill based on their historical learning trajectories. In recent years, a recurrent neural network model called deep knowledge tracing (*DKT*) has been proposed to handle the knowledge tracing task and literature has shown that *DKT* generally outperforms traditional methods [16], [17], [16]. Based on the use of a recurrent neural network, *DKT* is the first model that exhibited promising results using recurrent neural networks and suggested a promising new line of research for knowledge tracing (KT) in deep learning [16]. Following the *DKT* model, there are increasing amounts of researches. Zhang et al. (2017) extended the *DKT* model to incorporated more features at the item-level including first response time, attempt count, and first action. After convert to categorical data, those features were represented as a sparse vector by one-hot encoding as inputs [15]. Then Auto-Encoder was applied to reduce the dimensionality of inputs to *DKT*. Chen et al. (2018) proposed to incorporate the information of KC structures into the *DKT* model to solve the problem of model evaluation inaccuracy caused by data sparsity, which specifically refers to considering the pre and post-relationship of KCs [18]. Minn et al. (2018) proposed combines student's learning ability into *DKT* [19]. K-means was used to clustering the students into a group with similar ability at each time interval first and then combine that information with *DKT*. Yang et al. (2018) designed an automatic system to embed the heterogeneous features implicitly and effectively into the original *DKT* model [20].

BKT is a highly constrained, structured model. As mentioned before, it assumes that the student's knowledge state is binary, that predicting performance on an exercise requiring a given skill depends only on the student's binary knowledge state, and that the skill associated with each exercise is known in advance. If correct, these assumptions allow the model to make strong inferences. If incorrect, they limit the model's performance. The only way to determine if model assumptions are correct is to construct an alternative model that makes different assumptions and to determine whether the alternative outperforms *BKT*. *DKT* is exactly this alternative model, and its strong performance directs us to examine *BKT*'s limitations [21].

These four different methods are applied on the given data set after some initial preprocessing. To compare the different model performance we used two metrics: "RMSE" (root-mean-square error) and "AUC" (Area Under the Curve).

III. Our approach

To answer the main question of this work we applied all the knowledge tracing models mentioned above on different data. We decided to split initial data set into two different sets. The first one was created to analyze knowledge by two different subjects (Math and German). We will name this data set *subjects*. The second one was build in the way to trace knowledge by different topics within two subjects. We will name this data set *topics*. To build these data sets we merged different initial files, left only users that appear in all the files we used. Finally, both of these data frames had next columns: *user_id, learn_session_id, transaction_id, skill_name, correct, prior_success* and *prior_failure*.

After we created two different data sets, we ran various experiments on each of them.

IV. Experiments

A. Subjects

For *subjects* data set we tried 5-steps preprocessing approach:

1) Ways of considering partial answers
2) Task types
3) Acceptance of session
4) Closeness of session
5) Activeness of users

On the first step we have decided how to consider *PARTIAL ANSWERS*: as *CORRECT* (1) or *INCORRECT* (0). The second step was connected with such parameter as *type of task*. In the given data there are only two main types of tasks: 'learn' and 'level check'. On this step we have filtered data by this value (if we consider all the tasks or only tasks that have type 'level check'). On the third step we had two options: consider only sessions which result users finally accepted or consider all the sessions. The fourth filter was about closeness of sessions: if we consider only sessions that were closed or any sessions. The last but not the least step was about users. We tried to take into consideration only users that are active (have a lot of sessions) or any users.

All these steps are graphically shown in the Fig.6. Each layer of the tree corresponds to one of the steps. Each node consists of information about the step and show the total amount of rows after this step was applied. Finally we got $2^5 = 32$ different sets and ran all the models on them measuring *AUC* and *RMSE* metrics. A table below the tree shows values of these metrics, while each column of the table corresponds to one set.

After we did all the experiments, we chose 29^{th} subset out of 32 and got predictions of learning curves on it obtained with *PFA* model for each of subjects to see how well models predict. The real learning curves, its predictions and number of opportunities below are shown on Fig.1 for Math and on Fig.2 for German, respectively.

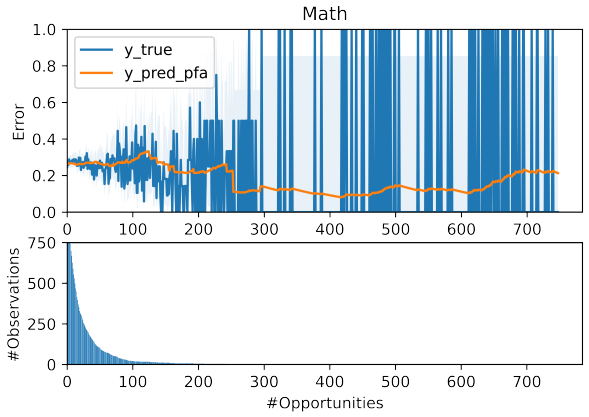


Fig. 1. Learning curves for Math

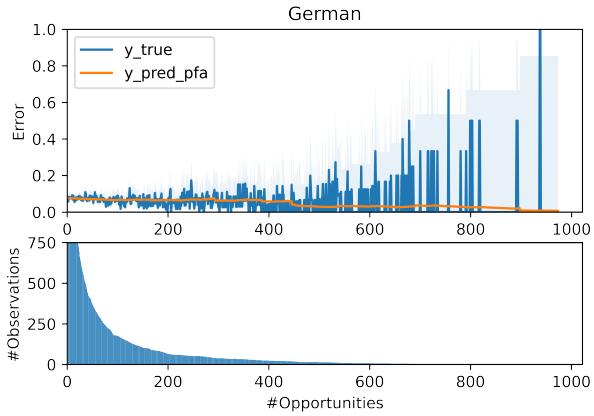


Fig. 2. Learning curves for German

B. Topics

For *topics* data set we performed other type of experiments to analyze how knowledge models performance depend on the popularity of different learning topics. We divided this data set into 10 difference subsets consisting of most and least 15, 18, 20, 22 and 25 popular among users topics. On each of these subsets we ran all the models again and measured the same metrics. Results of these experiments are shown in the Table.I.

Exp	BKT		AFM		PFA		DKT	
	AUC	RMSE	AUC	RMSE	AUC	RMSE	AUC	RMSE
Exp1	0.596	0.495	0.564	0.502	0.746	0.455	0.447	0.55
Exp2	0.657	0.482	0.602	0.496	0.762	0.445	0.555	0.516
Exp3	0.653	0.478	0.593	0.495	0.774	0.439	0.457	0.512
Exp4	0.688	0.466	0.655	0.475	0.796	0.426	0.646	0.492
Exp5	0.688	0.467	0.668	0.475	0.791	0.428	0.646	0.484
Exp6	0.652	0.48	0.606	0.489	0.72	0.464	0.593	0.499
Exp7	0.667	0.475	0.634	0.483	0.744	0.455	0.608	0.495
Exp8	0.646	0.481	0.603	0.49	0.731	0.461	0.584	0.503
Exp9	0.666	0.477	0.622	0.485	0.749	0.453	0.605	0.497
Exp10	0.673	0.474	0.634	0.483	0.751	0.452	0.6	0.5

TABLE I
RESULTS OF TOPICS EXPERIMENTS. EXPERIMENT 1 TO 5 ARE WITH CONSIDERING THE LEAST 15, 18, 20, 22, AND 25 POPULAR TOPICS AND EXPERIMENTS 6 TO 10 ARE CONSIDERING THE MOST 15, 18, 20, 22, AND 25 POPULAR TOPICS.

Another type of experiments we did was about time measuring for training all the models we used for *topics*. We averaged time spent on training models on subset of most and least popular topics. It is shown on Fig.3 with corresponding *AUC* and *RMSE* values. It should be mentioned that almost the same situation happened when we trained the model on *subjects* subsets. Hence, this figure explains the overall situation with time spent on training *BKT*, *AFM*, *PFA* and *DKT* on this data set.

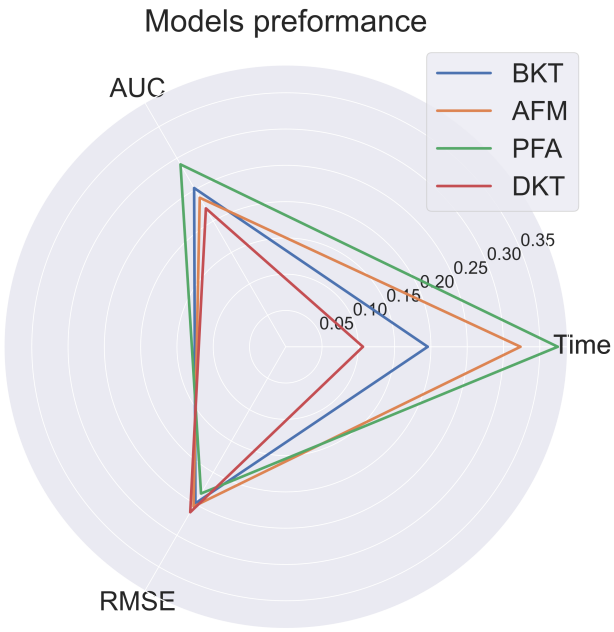


Fig. 3. Model performance by training time, AUC and RMSE

V. DISCUSSION

A. Subjects

To analyze the results of experiments performed on *subjects* data set, we need to explore the table below the tree Fig.6. The best results in each line are highlighted. As we can see, the biggest difference is between results we got for the first 16 subsets and the last 16 subsets. In other words, **how we consider partial answers is a criteria that influences the most models performance**. Other 'filters' are much less important and models based on the subsets from one of two sides of the tree have quite similar performance. Therefore, it is interesting to analyze left and right part of the tree independently. In average all the models based on subsets from the left part have performance that is shown in the next Fig.4.

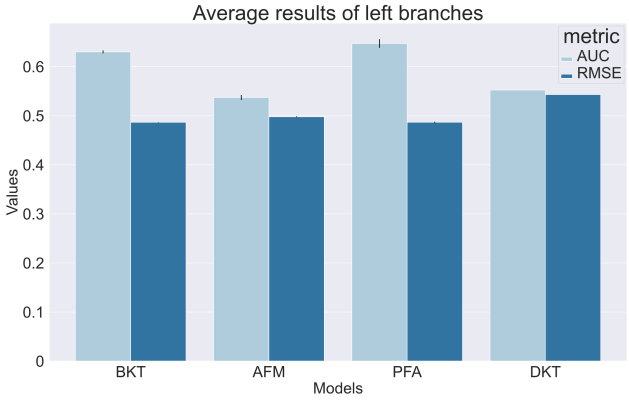


Fig. 4. Average performance of models from the left side

It can be observed that the *PFA* model exhibited a higher *AUC* score (around 0.66) and a lower standard deviation of *AUC* score across folds with respect to *AFM*, *BKT* and *DKT* models, indicating that the predictive power of the *PFA* model is higher and more stable across folds than the one of the other three models, when *AUC* is considered. Moreover, it is shown that *PFA* model performs better, on average, with respect to *AFM*, *BKT* and *DKT* in terms of *RMSE*, i.e. the *RMSE* score is lower for the *PFA* model. *BKT* model is on the second place by quality with *AUC* score not so far from the best one, whereas other two models have much lower quality on this data.

Similar situation we can observe in the Fig.5 that shows an average performance of models built on subsets from the right part of the tree.

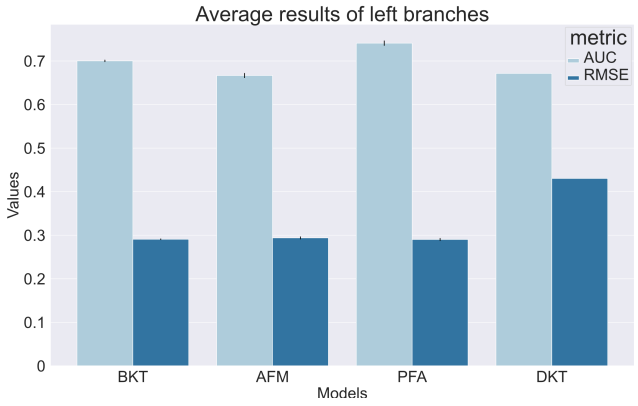


Fig. 5. Average performance of models from the right side

Again *PFA* model performs the best, while *BKT* is not so far from it and other two models are significantly worse. The main difference is that all the models have higher *AUC* scores and lower *RMSE* scores comparing to the corresponding results in the previous graph, but the difference between models performance became less. Moreover, *DKT* model has much higher *RMSE* score comparing to other models. This outperformance of *PFA* model can be explained by many reasons, for example, by the fact that the skills at hand seem to meet well *PFA* assumptions, specifically that (i) students may initially know more or less , (ii) students learn at the same rate, (iii) students learning rate differs for correct and wrong practice opportunities, (iv) each item belongs to one or more skills.

From the Fig.1 we can see the quality of predictions obtained by *PFA* model on the subset on which most of the models showed the best performance. From the error rate in the ground-truth data (y_{true}, blue), it can be observed that this skill appears not so easy for students, with an initial error rate of around 0.3 in the first opportunity. The error rate does not go down, as we would usually expect, it stays stable with some alternations for around 100 first opportunities and even go up a little bit. But after that it decreases slightly at around 150 opportunities. There is no reason to analyze its further behavior because the number of samples with such a big amount of opportunities is very low and does not show anything significant. Moreover, after 150-180 opportunities it becomes jumping from 0 to 1. Predictions we got describe all the behaviour very well with initial stability, a small increase and a fall near to 150 opportunities.

There is different situation in another Fig.2 that corresponds to real learning curves, its predictions obtained by *PFA* model and number of opportunities below. Based on the patterns of the ground-truth data (y_{true}, blue), the error rate observed for this skill at the earlier stages is of around 0.1. It means that most of users already have good knowledge. With number of opportunities it declines slightly and starts jumping from 0 to 1 when there is not enough of samples (at more than 400 number of opportunities). The predicted curve is just in the middle of the real one and perfectly fits it.

For both Math and German subjects the amount of students decreases slowly. Predictions obtained by *PFA* model are good and catch the main trend as far as there is enough data. The confidence interval for the real data increases with number of opportunities with Overall, ***PFA* model have an extremely good quality of predictions**.

B. Topics

In the Table.I, lines with the best models' performance are highlighted. These lines correspond to experiments ran on subsets containing information about least popular 22 and 25 topics, respectively. It is a very interesting result showing that for this kind of tasks and data we can reach good results even though using small amount of data.

In the Fig.3 we can see that in overall *PFA* model performs best, while it takes too much time to train comparing with other models. At the same time *BKT* model has performance not so far from the best model, whereas it spends significantly less time to train. *AFM* model spends a lot amount of time to train and shows very poor quality, while *DKT* model is the quickest one and the worst one by performance on this data.

VI. Conclusion

Among all the experiments and two different data sets *PFA* model performed the best and *BKT* is on the second place almost every time. It might be that *AFM* and *DKT* are much less qualified for this data. It should be mentioned that on datasets with more than 20000 rows *AFM* and *PFA* train for a much bigger amount of time than *BKT* and *DKT*. Therefore, there is always a tradeoff between time (*BKT*) and quality (*PFA*). Overall, working with this dataset and having enough amount of data we would advise to use *BKT* model as found out by [22], while for small datasets or when the quality of the model is crucially important *PFA* should be used.

REFERENCES

[1] Q. Liu, S. Shen, Z. Huang, E. Chen, and Y. Zheng, "A survey of knowledge tracing," *arXiv preprint arXiv:2105.15106*, 2021.

[2] M. Dai, J.-L. Hung, X. Du, H. Tang, and H. Li, "Knowledge tracing: A review of available technologies," *Journal of Educational Technology Development and Exchange (JETDE)*, vol. 14, no. 2, p. 1, 2021.

[3] C. Gabriella, L. Grilli, P. Limone, S. Domenico, and S. Daniele, "Deep learning for knowledge tracing in learning analytics: an overview," in *First Workshop on Technology Enhanced Learning Environments for Blended Education-The Italian e-Learning Conference 2021*, vol. 2817. CEUR-WS, 2021, pp. 1–10.

[4] L. St.Gallen, "Lerne deutsch und mathematik fürs gymnasium oder die fachmittelschule. lernnavi erfasst deinen lernstand, stellt dir dazu passende aufgaben zusammen und gibt dir bei jeder aufgabe ein feedback." [Online]. Available: https://www.lernnavi.ch/

[5] Cahlr, "Cahlr/pybkt: Python implementation of bayesian knowledge tracing and extensions." [Online]. Available: https://github.com/CAHLR/pyBKT

[6] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized bayesian knowledge tracing models," in *International conference on artificial intelligence in education*. Springer, 2013, pp. 171–180.

[7] S. Doroudi and E. Brunskill, "Fairer but not fair enough on the equitability of knowledge tracing," in *Proceedings of the 9th international conference on learning analytics & knowledge*, 2019, pp. 335–339.

[8] J.-J. Vie and H. Kashima, "Knowledge tracing machines: Factorization machines for knowledge tracing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 750–757.

[9] W. Gan, Y. Sun, X. Peng, and Y. Sun, "Modeling learner's dynamic knowledge construction procedure and cognitive item difficulty for knowledge tracing," *Applied Intelligence*, vol. 50, no. 11, pp. 3894–3912, 2020.

[10] M. Zhang, X. Zhu, C. Zhang, Y. Ji, F. Pan, and C. Yin, "Multi-factors aware dual-attentional knowledge tracing," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2588–2597.

[11] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger, "Performance factors analysis–a new alternative to knowledge tracing." *Online Submission*, 2009.

[12] Y. Gong, J. E. Beck, and N. T. Heffernan, "Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures," in *International conference on intelligent tutoring systems*. Springer, 2010, pp. 35–44.

[13] ——, "How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis," *International Journal of Artificial Intelligence in Education*, vol. 21, no. 1-2, pp. 27–46, 2011.

[14] J. González-Brenes, Y. Huang, and P. Brusilovsky, "General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge," in *The 7th international conference on educational data mining*. University of Pittsburgh, 2014, pp. 84–91.

[15] 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*, 2017, pp. 169–172.

[16] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," *Advances in neural information processing systems*, vol. 28, 2015.

[17] C.-K. Yeung, "Deep-irt: Make deep learning based knowledge tracing explainable using item response theory," *arXiv preprint arXiv:1904.11738*, 2019.

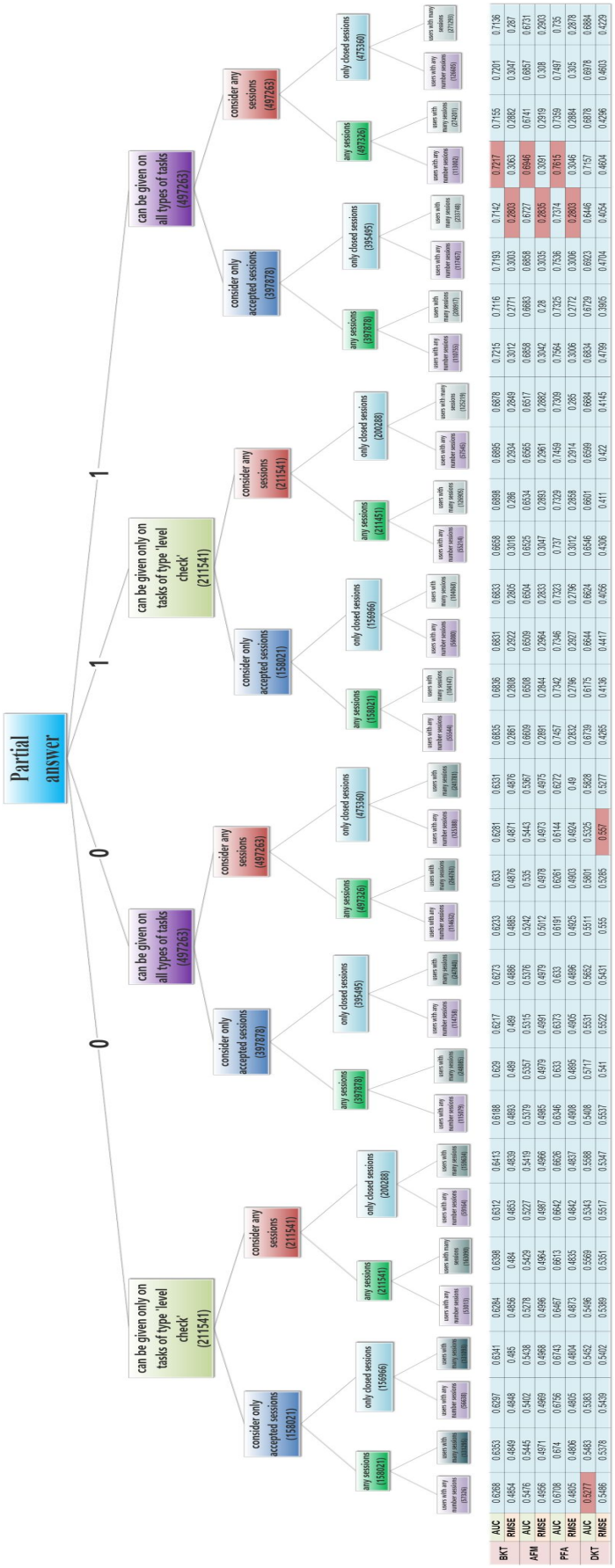
[18] P. Chen, Y. Lu, V. W. Zheng, and Y. Pian, "Prerequisite-driven deep knowledge tracing," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 39–48.

[19] S. Minn, Y. Yu, M. C. Desmarais, F. Zhu, and J.-J. Vie, "Deep knowledge tracing and dynamic student classification for knowledge tracing," in *2018 IEEE International conference on data mining (ICDM)*. IEEE, 2018, pp. 1182–1187.

[20] H. Yang and L. P. Cheung, "Implicit heterogeneous features embedding in deep knowledge tracing," *Cognitive Computation*, vol. 10, no. 1, pp. 3–14, 2018.

[21] M. Khajah, R. V. Lindsey, and M. C. Mozer, "How deep is knowledge tracing?" *arXiv preprint arXiv:1604.02416*, 2016.

[22] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell *et al.*, "When is deep learning the best approach to knowledge tracing?" *Journal of Educational Data Mining*, vol. 12, no. 3, pp. 31–54, 2020.



BKT	AUC	0.6288	0.6353	0.6297	0.6341	0.6284	0.6398	0.6312	0.6413	0.6188	0.629	0.6217	0.6273	0.6233	0.633	0.6281	0.6331	0.6835	0.6836	0.6831	0.6833	0.6658	0.6898	0.6895	0.6878	0.7215	0.7116	0.7193	0.7142	0.7217	0.7155	0.7201	0.7138
	RMSE	0.4654	0.4849	0.4848	0.485	0.4656	0.484	0.4853	0.4839	0.4863	0.489	0.489	0.4866	0.4885	0.4876	0.4871	0.4876	0.4861	0.2868	0.2822	0.2805	0.3018	0.286	0.2934	0.2849	0.3012	0.2771	0.3003	0.2803	0.3063	0.2862	0.3047	0.287
AFM	AUC	0.5476	0.5445	0.5402	0.5438	0.5278	0.5429	0.5227	0.5419	0.5379	0.5357	0.5315	0.5376	0.5242	0.535	0.5443	0.5367	0.6609	0.6508	0.6539	0.6504	0.6525	0.6534	0.6565	0.6517	0.6558	0.6683	0.6658	0.6727	0.6946	0.6741	0.6857	0.6731
	RMSE	0.4956	0.4971	0.4869	0.4968	0.4696	0.4964	0.4987	0.4966	0.4985	0.4979	0.4891	0.4979	0.5012	0.4978	0.4973	0.4975	0.2891	0.2844	0.2964	0.2833	0.3047	0.2893	0.2961	0.2882	0.3042	0.28	0.3035	0.2835	0.3091	0.2919	0.308	0.2903
PFA	AUC	0.6708	0.674	0.6756	0.6743	0.6467	0.6613	0.6642	0.6626	0.6346	0.633	0.6373	0.633	0.6191	0.6261	0.6144	0.6272	0.7457	0.7442	0.7346	0.7323	0.737	0.7329	0.7459	0.7309	0.7664	0.7325	0.7536	0.7374	0.7615	0.7359	0.7497	0.735
	RMSE	0.4805	0.4806	0.4805	0.4804	0.4873	0.4835	0.4842	0.4837	0.4908	0.4895	0.4905	0.4866	0.4925	0.4903	0.4924	0.49	0.2832	0.2786	0.2927	0.2796	0.3012	0.2858	0.2914	0.285	0.3006	0.2772	0.3006	0.2803	0.3046	0.2884	0.305	0.2878
DKT	AUC	0.5277	0.5453	0.5383	0.5452	0.5496	0.5369	0.5343	0.5388	0.5408	0.5717	0.5531	0.5652	0.5511	0.5601	0.5525	0.5628	0.6739	0.6715	0.6644	0.6624	0.6546	0.6601	0.6599	0.6694	0.6834	0.6729	0.6823	0.6446	0.7157	0.6878	0.6978	0.6864
	RMSE	0.5486	0.5378	0.5439	0.5402	0.5389	0.5351	0.5517	0.5347	0.5537	0.541	0.5522	0.5431	0.555	0.5285	0.557	0.5277	0.4265	0.4136	0.4417	0.4056	0.4306	0.411	0.422	0.4145	0.4799	0.3905	0.4704	0.4054	0.4604	0.4296	0.4603	0.4229