



# An XGBoost-Based Knowledge Tracing Model

Wei Su<sup>1,2</sup> · Fan Jiang<sup>3</sup> · Chunyan Shi<sup>4</sup> · Dongqing Wu<sup>5</sup> · Lei Liu<sup>4</sup> · Shihua Li<sup>1</sup> · Yongna Yuan<sup>1</sup> · Juntai Shi<sup>1</sup>

Received: 21 August 2021 / Accepted: 30 January 2023  
© The Author(s) 2023

## Abstract

The knowledge tracing (KT) model is an effective means to realize the personalization of online education using artificial intelligence methods. It can accurately evaluate the learning state of students and conduct personalized instruction according to the characteristics of different students. However, the current knowledge tracing models still have problems of inaccurate prediction results and poor features utilization. The study applies XGBoost algorithm to knowledge tracing model to improve the prediction performance. In addition, the model also effectively handles the multi-skill problem in the knowledge tracing model by adding the features of problem and knowledge skills. Experimental results show that the best AUC value of the XGBoost-based knowledge tracing model can reach 0.9855 using multiple features. Furthermore, compared with previous knowledge tracing models used deep learning, the model saves more training time.

**Keywords** Knowledge tracing · Deep learning · Multi-skill · XGBoost

## Abbreviations

KT	Knowledge tracing
MOOC	Massive open online course
ITSs	Intelligent tutoring systems
LPP	Learner performance prediction
IRT	Item response theory
DINA	Deterministic inputs, noisy “and” gate
MIRT	Multidimensional item response theory
MF	Matrix factorization
BKT	Bayesian knowledge tracing
DKT	Deep knowledge tracing
LSTM	Long-short term memory
KC	Knowledge component
DKVMN	Dynamic key-value storage network
SAKT	The self-attentive knowledge tracing
XGBoost	EXtreme gradient boosting
AutoInt	Automatic feature interaction

FM	Factorization machines
DeepFM	Deep factorization machines
AUC	Area under curve
ACC	Accuracy

## 1 Introduction

The rapid growth of educational resources on the Internet has brought lots of popular online learning platforms, such as massive open online course (MOOC) platforms [1] and intelligent tutoring systems (ITSs) [2]. Some ITS can trace the current students' mastery of knowledge through the performance of answering questions in the past [3, 4]. So students can obtain appropriate guidance in the process of acquiring relevant knowledge [5, 6]. The development of artificial intelligence technology makes these online learning platforms more and more intelligent, and brings a very wide range of advantages [7, 8]. For example, these platforms can automatically provide personalized feedback and learning suggestions to each student by analyzing the historical learning data of each student [9, 10]. Behind these personalized tutoring services, the learner performance prediction technology (LPP) plays a key role. The technology predicts future practice performance of the learners through their proficiency in mastering skills and concepts.

In fact, the knowledge proficiency of learners will change over time, which is the result of learners acquiring and

✉ Wei Su  
suwei@lzu.edu.cn

<sup>1</sup> School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu, China  
<sup>2</sup> Key Laboratory of Media Convergence Technology and Communication, Lanzhou, Gansu, China  
<sup>3</sup> CITIC Bank Software Development Center, Beijing, China  
<sup>4</sup> Duzhe Publishing Group Co. Ltd., Lanzhou, Gansu, China  
<sup>5</sup> Yizhichuan Primary School, Lanzhou, Gansu, China

forgetting knowledge [11, 12]. Hence the task of LPP should be based on learners' dynamic knowledge states implicitly contained in their learning logs, where knowledge tracing comes into play. A key problem in learners' data analysis is to predict learners' future performance, given their past performance, which is referred to as the knowledge tracing problem [13, 14].

Usually, the knowledge tracing task can be described as: given a specific learning task, we predict the learner's next performance  $x_{t+1}$  by observing the sequence  $x = \{x_1, \dots, x_t\}$  of the learner's historical performance on a specific problem. A common task in knowledge tracing is feature selection,  $x_t$  can be represented by a tuple  $(q_t, a_t)$ , where  $q_t$  and  $a_t$  are, respectively, the question information of the learner's answering to at the time  $t$  and the right or wrong of the answer, so the probability  $P(a_{t+1} = 1 | q_{t+1}, x_t)$  of the learner answering the question correctly at the time  $t + 1$  can be predicted [15, 16].

The existing knowledge tracing models still have some problems such as low prediction accurate, and hard to predict for multi-skills problem [17]. Therefore, the further research is necessary to improve the performance of the model and solve the multi-skill problem. The study which is applying XGBoost algorithm to knowledge tracing tasks improves both the performance and efficiency of the knowledge tracing model.

## 2 Related Work

The knowledge tracing (KT) model, first proposed by Atkinson, is a classic model for diagnosing and tracing learners' learning states [18]. The existed knowledge tracing models can be divided into two types: statistical KT models and deep learning KT models.

For statistical KT models, Bayesian Knowledge Tracing (BKT) model is one of the most classical knowledge tracing models [19]. BKT models the learners' latent knowledge state as a set of binary variables, representing whether to master a certain knowledge component (KC). After each learned interaction, BKT uses a hidden Markov model (HMM) to update the probabilities of these binary variables [20, 22].

As deep learning develops, many researchers have applied it into the field of knowledge tracing. The deep knowledge tracing (DKT) model [23] was proposed in 2015, and its basic structure is recurrent neural network (RNN). Inspired by the concepts of Memory-augmented neural networks (MANN), the idea of augmenting dynamic key-value memory networks (DKVMN) [24] with an auxiliary memory was proposed. DKVMN explicitly maintains a KC representation matrix (key) and a knowledge state representation matrix (value). Sequential Key-Value Memory Network (SKVMN)

[25] has combined the strengths of DKT's recurrent modeling capacity and DKVMN's memory capacity for knowledge tracing. For text-aware knowledge tracing models, the Exercise-enhanced RNN (EERNN) [26] uses a bi-directional LSTM module to extract the representation of each question from the question's text. The Exercise-aware Knowledge Tracing (EKT) [27] model, integrated the EERNN model and the DKVMN model, has a dual attention module. A lot of attempts have been made to use attention mechanism to enhance model interpretability. Pandey et al. [28] took the lead in using the Transformer model in the field of knowledge tracing and proposed the SAKT model. Choi et al. [29] proposed a model to improve self-attentive computation for knowledge tracing adaptation and named the Separated Self-attentive Neural Knowledge Tracing (SAINT). In addition, further work has been made through using other deep learning models for knowledge tracing. The Graph-Based Knowledge Tracing (GKT) [30] model attempts to incorporate a graph where nodes represent KCs and edges represent the dependency relation between KCs for a relational inductive bias. A joint graph convolutional network-based Deep Knowledge Tracing (JKT) [31] framework is proposed to model the multi-dimensional relationships into graph. The Convolutional Knowledge Tracing (CKT) [32] model is the first model to use convolutional neural networks in the field of knowledge tracing. A neural Turing machine-based skill-aware knowledge tracing (NSKT) [33] for conjunctive skills was proposed, which can capture the relevance among the knowledge concepts of a question to model students' knowledge state more accurately and to discover more latent relevance among knowledge concepts effectively. For various DKT models, they have higher prediction performance than most of statistical KT models, but they still have the following problems: first, most DKT models only use few features and do not try different combination of features; secondly, due to over-parameterized black-box nature of deep learning, it is often difficult to understand the prediction results of DKT; finally, most DKT models need more training time and they only have good prediction result on huge data amount.

## 3 Knowledge Tracing Model Based on XGBoost

The XGBoost (eXtreme Gradient Boosting) algorithm [34, 35] is a Boosting-type integrated learning algorithm, which completes the learning task by constructing and combining multiple weak learners. The basic idea of the XGBoost algorithm is to continuously add different trees to the model, and to make the tree model grow through feature splitting. Each time adding a tree is equivalent to learning a new function, and then fitting the residual of the last prediction. Finally, the predicted value of the sample is the sum of the scores

of all trees on the sample. Combining the characteristics of knowledge tracing tasks and the advantages of strong generalization ability and high computational efficiency of XGBoost, this study constructs a knowledge tracing model based on the XGBoost algorithm [36]. The flow chart of the model is shown in Fig. 1.

The input data of XGBoost model can be different combination features of online learning platform including question, knowledge skill, attempt time, students answer, etc. Assuming that the dataset contains N samples and M-dimensional features, the model containing k decision trees can be represented by  $\hat{y}_i$ . The calculation is shown in formula (1):

$$\hat{y}_i^{(t)} = \sum_{k=1}^t f_k(x_i) \tag{1}$$

The  $t - th$  round of model prediction in the decision tree is shown in formula (2):

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{2}$$

where  $\hat{y}_i^{(t-1)}$  represents the predicted value of round  $t - 1$ , and  $f_t(x_i)$  is the score of the sample at round  $t$ . The objective function calculation of XGBoost model is shown as formula (3):

$$Obj^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + C \tag{3}$$

The  $t - th$  round of regularization is obtained by the sum of the regularization terms of all trees. The objective function

consists of two parts: one is the difference between the real value and the predicted value, and the other part is the regularization function to prevent overfitting during the training of XGBoost model. The regularization function consists of two parts, as shown in Eq. (4):

$$\Omega(f_t) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^n w_j^2 \tag{4}$$

where  $T$  refers to the number of leaf nodes,  $\gamma$  and  $\lambda$  represent the penalty coefficient, and  $w_j$  represents the score of the  $j - th$  leaf node. The expansion of  $Obj$ 's second-order Taylor formula is shown in Eq. (5):

$$Obj^{(t)} = \sum_{i=1}^n \left[ l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] \tag{5}$$

Among them,  $g_i$  refers to the first-order partial derivative of the loss function of the  $i - th$  sample, and  $h_i$  refers to the second-order partial derivative of the loss function of the  $i - th$  sample. When the model is trained, its objective function is shown in formula (6):

$$Obj^{(t)} \approx \sum_{j=1}^T \left[ \left( \sum g_i \right) w_j + \frac{1}{2} \left( \sum h_i + \lambda \right) w_j^2 \right] + \gamma T \tag{6}$$

Among them,  $obj$  is a quadratic equation of one variable about  $w_j$ , and the value of  $obj$  is the smallest when  $w_j = -G_j / (H_j + \lambda)$ . At this time, its objective function is shown in formula (7):

$$Obj^{(t)} = \frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T \tag{7}$$

Among them,  $G_j = \sum g_i$ ,  $H_j = \sum h_i$  and  $obj$  is equivalent to the function of the Gini coefficient, which evaluates the quality of the tree.

The XGBoost algorithm selects the feature with the largest score as the split feature. Through the above analysis, the XGBoost algorithm can predict the probability of students doing the right question based on student data. Therefore, the XGBoost model can be applied to knowledge tracing tasks. The model selects relevant student's characteristic data as input, such as student id, knowledge skill, question id, the number of prompts, the number of attempts to question, etc. In the next section, experiments will be conducted to verify the effectiveness of the XGBoost knowledge tracing model.

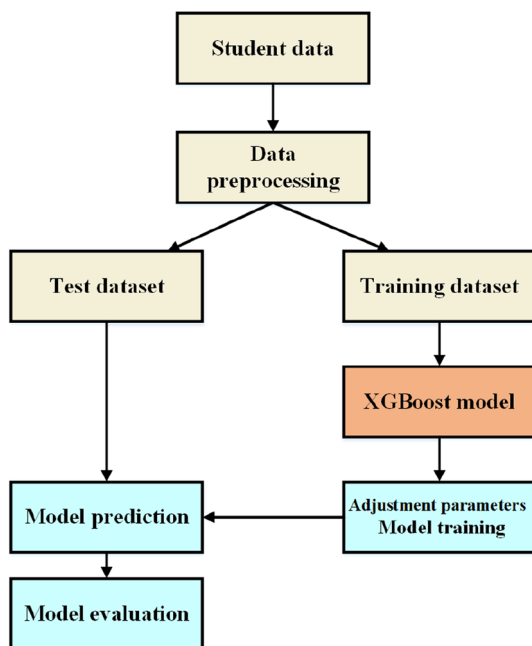


Fig. 1 Flow chart of XGBoost knowledge tracing model

**Table 1** The statistics of the three datasets

Dataset	Students	Skills	Interactions
ASSIST09	4217	124	525,535
Algebra08	3310	922	8,918,054
ASSIST17	686	102	942,816

## 4 Dataset

### 4.1 Dataset Description

The experiment uses three datasets, including ASSIST09, Algebra08 and ASSIST17. The number of students, knowledge skills and interactions in the three datasets are shown in Table 1. An interaction is an answering record for a question by a student.

ASSIST09 and ASSIST17 are student answer data collected from the ASSISTments online teaching platform 2009 and 2017. The dataset of ASSIST09 contains a total of 525,535 interactions for 124 knowledge skills by 4,217 students. The dataset of ASSIST17 contains 942,816 interactions, 686 students and 102 knowledge skills.

Algebra08 [37] has records of 2008–2009 interactions between students and computer-aided-tutoring systems. In this dataset, there are 8,918,054 interactions of 3,310 students on 922 knowledge skills.

### 4.2 Data Preprocessing

Since the datasets come from online teaching platforms, there is a certain amount of noisy data in the dataset, including missing values, duplicate values, outliers, etc. A pre-process for data cleaning is necessary before feeding the data into the model. The steps of data preprocessing in our work include outlier processing, missing value processing and data labeling. Outlier data are extreme values that deviate from other observations on dataset, they may indicate a variability in a measurement, experimental errors or a novelty. For example, there are some minus value (such as '-7,759,575') for overlap\_time column in ASSISTments data. Obviously, it is not a time value and should be deleted. For Missing value processing, the skill\_id feature is knowledge skill ID which a question belongs to, but there are 66,326 empty values which should be deleted.

## 5 Model Training

The training of model uses the scikit-learn library in Python, and the main parameters of XGBoost are shown in Table 2.

**Table 2** The main parameters of XGBoost model

Parameter name	Meaning	Default	Range
Eta	Learning rate	0.3	[0, 1]
Gamma	Decline of the loss function	0	[0, ∞]
Max_depth	Maximum depth of the tree	6	[0, ∞]
Min_child_weight	The weight of the smallest leaf node	1	[0, ∞]
Subsample	Random sampling	1	[0, ∞]
Colsample_bytree	Random number of columns	1	[0, ∞]
Lambda	L2 regularization	1	–
Alpha	L1 regularization	0	–
n_estimators	Training times	100	[1, ∞]

**Table 3** The parameters of FM model

Name	Paraphrase	Value
Num_iter	Number of iterations	100
Learning_method	Study method	mcmc
Dimension	Dimension	5

**Table 4** The parameters of deepFM model

Name	Paraphrase	Value
Learning_rate	Learning rate	5e– 6
Epsilon	–	1e– 10
Dropout	Loss rate	0.5
l2_reg	L2 regularization	1e– 2
Batch_size	Number of batches	256
Epochs	Training times	200

### 5.1 Model Evaluation

In the knowledge tracing task, *Logloss* is used as the loss function in the experiment, and its calculation is shown in Eq. (8) [38].

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y_i \log p(x_i) + (1 - y_i) \log (1 - p(x_i))] \quad (8)$$

where  $y_i$  is the true category of instance  $x_i$ ,  $p(x_i)$  is the probability of predicting input instance  $x_i$  correctly.

### 5.2 Parameters

In the paper, the knowledge tracing models based on FM, DeepFM, AutoInt, XGBoost and DKT are chosen as the baseline models [39]. The parameters of the FM, DeepFM,

**Table 5** The parameters of autoint model

Name	Paraphrase	Value
Learning_rate	Learning rate	5e- 5
Epsilon	-	1e- 10
Dropout	loss rate	0.5
l2_reg	L2 regularization	1e- 1
Batch_size	Number of batches	256
Epochs	Training times	200

**Table 6** The parameters of XGBoost model

Name	Value
Eta	0.3
Gamma	0.2
Max_depth	9
Min_child_weight	1
Subsample	0.9
Colsample_bytree	0.9
Lambda	1e- 5
Alpha	0.1
n_estimators	300

**Table 7** The parameters of DKT model

Name	Paraphrase	Value
Learning_rate	Learning rate	1e- 3
Hidden_size	Number of hidden layer units	200
Hidden_layer_num	Number of hidden layers	1
Batch_size	Number of batches	64
Epochs	Training times	200

AutoInt, XGBoost and DKT model are shown in Tables 3, 4, 5, 6 and 7, respectively.

Among the models, the FM model and AutoInt model respectively apply the FM and AutoInt algorithm to the knowledge tracing task; the DeepFM model includes two parts: the FM model and the feedforward neural network.

## 6 Experimental Results and Analysis

In the experiment, FM, DeepFM, AutoInt, and XGBoost models use feature combinations for model training and predict the results. The DKT, which is regression prediction model, uses time series data constructed with user\_id and skill\_id features to predict the results. We use the area under the receiver operating characteristic curve (AUC) as an evaluation metric to compare prediction performance among the models. A higher AUC indicates better performance. For the knowledge tracing model based

on FM, DeepFM, AutoInt and XGBoost, the correct or wrong answer of a student is regarded as a label and so it is treated as a classification problem. The result of the model is the probability of students answering knowledge skills. It is also reasonable to use AUC as the indicator of the model. The time recorded in the experiments is the total training time. The running time of the FM model is affected with the num\_iter parameter, and the running time of the XGBoost model is affected with the n\_estimators parameter. The running time of the DeepFM, AutoInt and DKT models is influenced by epochs and batch\_size. In addition, the models include DeepFM, AutoInt and DKT run on GPU machines, while the other models based on FM and XGBoost run on CPU machines [40].

Tables 8, 9 and 10 show the experimental results of the knowledge tracing models based on FM, DeepFM, AutoInt, XGBoost and DKT for the ASSIST09, Algebra08 and ASSIST17 datasets using different features. We also give the bar charts of the experimental results for a visual representation in Figs. 2 and 3 for the ASSIST09 dataset.

From Tables 8 and 9, when only the user\_id and skill\_id features are used, the XGBoost model has no better performance than the AutoInt model. After adding the feature of problem\_id, from Tables 8 and 10, the performance of the XGBoost model is better than the models based on FM, DeepFM, and AutoInt. But for the Algebra08 dataset, the XGBoost model does not perform better than other models. When the model adds the features of attempt\_count and extra, for ASSIST09 dataset, the XGBoost model can reach 0.9855 (AUC) and 0.9442 (ACC). Compared with the basic model using user\_id and skill\_id, its AUC is increased by 0.234 and ACC is increased by 0.204. However, from Table 10, for the ASSIST17 dataset, when using all features, the AUC is only 0.7860.

Compared with all other models, the running time of the XGBoost model is minimum. When using user\_id and skill\_id features, although the models based on AutoInt and DKT can perform better, the training time of the models is much longer than the XGboost model. It is not conducive to the model deployment and application of online education platforms. However, the knowledge tracing model based on the XGBoost algorithm can greatly reduce the training time of the model if it is deployed to online platforms.

From Tables 8 and 9, it can be found that the attempt\_count feature has a greater impact on the performance of the model, especially for Algebra08, adding the attempt\_count feature to the model can bring an improvement of 0.2963 (AUC) and 0.1221 (ACC) to the XGBoost model. For ASSIST09 and ASSIST17 datasets, if the extra feature is added to the model, it can bring an improvement to the XGBoost model. Therefore, the attempt\_count feature still has a greater impact on the prediction performance of the knowledge tracing model based on XGBoost.

**Table 8** The experimental results for ASSIST09

Feature	Model	AUC	ACC	Time
Feature 1: user, problem, skill, attempt, extra <sup>1</sup>	XGBoost	<b>0.9855</b>	<b>0.9442</b>	<b>0:00:41</b>
	AutoInt	0.9843	0.9412	1:29:52
	DeepFM	0.9818	0.9391	0:35:12
	FM	0.9256	0.9347	0:01:45
Feature 2: user, problem, skill, attempt	XGBoost	<b>0.9658</b>	<b>0.9173</b>	<b>0:00:39</b>
	AutoInt	0.9562	0.9055	1:04:49
	DeepFM	0.9577	0.9027	0:27:15
	FM	0.8925	0.8919	0:01:16
Feature 3: user, problem, skill, extra <sup>1</sup>	XGBoost	<b>0.8958</b>	<b>0.8405</b>	<b>0:00:38</b>
	AutoInt	0.8610	0.8059	1:20:59
	DeepFM	0.8593	0.8043	0:29:37
	FM	0.6977	0.7602	0:01:36
Feature 4: user, problem, skill	XGBoost	<b>0.8682</b>	<b>0.8111</b>	<b>0:00:37</b>
	AutoInt	0.8378	0.7796	0:59:22
	DeepFM	0.8105	0.7582	0:27:27
	FM	0.6539	0.7042	0:01:04
Feature 5: user, skill	DKT	<b>0.8583</b>	<b>0.8034</b>	0:10:25
	XGBoost	0.7513	0.7407	<b>0:00:28</b>
	AutoInt	0.7628	0.7444	0:53:53
	DeepFM	0.7489	0.7306	0:25:24
	FM	0.6505	0.7054	0:00:50

Bold fonts indicate the best results of the experiment

<sup>1</sup>Extra: combined features, including: first\_action, school\_id, teacher\_id, tutor\_mode

**Table 9** The experimental results for Algebra08

Feature	Model	AUC	ACC	Time
Feature 1: user, problem, skill, attempt	XGBoost	0.9547	<b>0.9394</b>	<b>0:00:15</b>
	AutoInt	0.9307	0.9129	0:20:30
	DeepFM	<b>0.9624</b>	0.9212	0:18:54
	FM	0.9547	0.9206	0:18:02
Feature 2: user, problem, skill	XGBoost	0.6584	0.8173	<b>0:00:15</b>
	AutoInt	0.7306	0.7945	0:22:15
	DeepFM	<b>0.7754</b>	<b>0.8220</b>	0:18:39
	FM	0.6612	0.8127	0:18:52
Feature 3: user, skill	DKT	<b>0.7920</b>	<b>0.8506</b>	1:35:04
	XGBoost	0.6181	0.8106	<b>0:00:13</b>
	AutoInt	0.7524	0.8243	0:20:44
	DeepFM	0.7612	0.8273	0:20:33
	FM	0.6181	0.8015	0:17:50

Bold fonts indicate the best results of the experiment

In addition, adding the problem\_id and skill\_id features to the XGBoost model can also accurately understand which problem or knowledge skill the model has mastered. It can

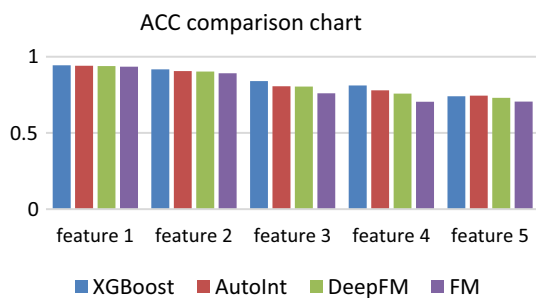
effectively deal with multi-skills problems without processing the original dataset.



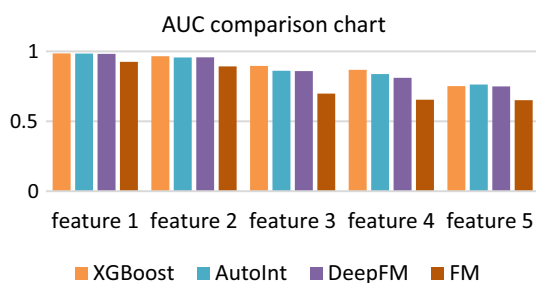
**Table 10** The experimental results for ASSIST17

Feature	Model	AUC	ACC	Time
Feature 1: user, problem, skill, attempt, extra <sup>1</sup>	XGBoost	0.7860	<b>0.7286</b>	<b>0:07:26</b>
	AutoInt	<b>0.7873</b>	0.7275	0:32:54
	DeepFM	0.7865	0.7283	0:15:02
	FM	0.7408	0.6976	0:18:45
Feature 2: user, problem, skill, attempt	XGBoost	0.7790	0.7210	<b>0:08:24</b>
	AutoInt	0.7788	0.7243	0:34:39
	DeepFM	<b>0.7799</b>	<b>0.7263</b>	0:16:02
	FM	0.7376	0.6977	0:18:51
Feature 3: user, problem, skill, extra <sup>1</sup>	XGBoost	<b>0.7748</b>	<b>0.7849</b>	<b>0:05:10</b>
	AutoInt	0.7421	0.7071	0:22:37
	DeepFM	0.7441	0.7110	0:18:32
	FM	0.7207	0.6946	1:52:42
Feature 4: user, problem, skill	XGBoost	<b>0.7841</b>	<b>0.7255</b>	<b>0:08:01</b>
	AutoInt	0.7128	0.6979	0:33:53
	DeepFM	0.7259	0.7042	0:16:34
	FM	0.7172	0.6981	0:16:32
Feature 5: user, skill	DKT	<b>0.6703</b>	0.6306	1:15:06
	XGBoost	0.6178	0.6492	<b>0:00:29</b>
	AutoInt	0.6508	0.6621	0:18:43
	DeepFM	0.6523	<b>0.6649</b>	0:17:33
	FM	0.6499	0.6596	1:49:25

Bold fonts indicate the best results of the experiment



**Fig. 2** ACC for XGBoost, AutoInt, DeepFM, and FM model



**Fig. 3** AUC for XGBoost, AutoInt, DeepFM, and FM models

## 7 XGBoost Model Analysis

It can be seen from above that using the XGBoost model for the knowledge tracing task has good experimental results. Using XGBoost model for knowledge tracing task also has the following advantages. Adding the complexity of the tree as a regularization term to the optimized target reduces the risk of overfitting. The work using the XGBoost model for the knowledge tracing task can parallelize the calculation between trees, which makes the calculation speed of the prediction stage faster.

Intersecting with other knowledge tracing models, especially various DKT models, using the XGBoost model saves more time. At the same time, after one iteration, XGBoost model will multiply the weight of the leaf node by the coefficient, mainly to weaken the influence of each tree, so that there is more space for learning later. Weakening the influence of previous tree can be used to represent the forgetting behavior of students during the learning process. For example, in the Attentive Knowledge Tracing (AKT) model, weight decay is used to account for forgetting effects in student memory over time by reducing the attentional weight of questions over a series of interactions. The XGBoost model can use machine learning method to achieve this function. XGBoost algorithm internally implements a boosted tree model, which can automatically handle missing values. However, when the amount

of training data in the knowledge tracing dataset is too large, and there is a suitable deep knowledge tracing model, the accuracy of deep learning can be far ahead of XGBoost.

## 8 Conclusion

The study introduces the basic principles of the XGBoost algorithm, and then the method of applying XGBoost into the knowledge tracing model. Experimental results show that XGBoost algorithm can be effectively applied to knowledge tracing tasks. When multiple features are added to the model, the best predicted AUC value can reach 0.9855. At the same time, compared with previous knowledge tracing models using deep learning, the XGBoost model saves more training time. It can be more conveniently deployed and applied on online education platforms.

**Author Contributions** Algorithm designing and analysis, WS; Algorithm programming and experiment implementation, FJ, JS; Paper writing, CS, JS, LL, SL; Revising and reviewing, WS, DW, YYuan.

**Funding** This work was supported by the Talent Innovation and Entrepreneurship Fund of Lanzhou (2020-RC-13), the Science and Technology Project of Gansu (21YF5GA102, 21YF5GA006, 22ZD6GA02) and the Key Laboratory of Media convergence Technology and Communication, Gansu Province (21ZD8RA008).

**Data Availability** The experimental data use the ASSIST09, Algebra08 and ASSIST17 datasets. The datasets are located at the following URL: <https://sites.google.com/site/assistmentsdata/home/assistance-2009-2010-data>, <http://pslclatashop.web.cmu.edu/KDDCup/downloads.jsp> and <https://sites.google.com/view/assistmentsdatamining/dataset/>.

**Code Availability** The code of the paper can be found on <https://github.com/lzuie2022/2022>.

## Declarations

**Conflict of Interest** The authors declare no conflict of interest.

**Ethics Approval** Not applicable.

**Consent to Participate** Not applicable.

**Consent for Publication** Not applicable.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will

need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Reich, J.: Rebooting MOOC research. *Science*. **347**(6217), 34–35 (2015)
2. Anderson, J.R.: Cognitive modelling and intelligent tutoring. *Artif Intell.* **42**(1), 7–49 (1990)
3. Wang, T., Xiao, B., Ma, W.: Student behavior data analysis based on association rule mining. *Int. J. Comput. Intell. Sys.* **15**(1), 1–9 (2022)
4. Koedinger, K.R., et al.: New potentials for data-driven intelligent tutoring system development and optimization. *AI Magaz.* **34**, 27–41 (2013)
5. Liu, S., et al.: An Intelligent question answering system of the liao dynasty based on knowledge graph. *Int. J. Comput. Intellig. Sys.* **14**, 1–12 (2021)
6. Piech, C., et al.: "Learning program embeddings to propagate feedback on student code. In: International conference on machine Learning. PMLR (2015)
7. Hasanov, A., Laine, T.H., Chung, T.-S.: A survey of adaptive context-aware learning environments. *J. Ambient Intelligence Smart Environ.* **11**(5), 403–428 (2019)
8. Atkinson, R.C.: Optimizing the learning of a second-language vocabulary. *J. Exp. Psychol.* **96**, 124 (1972)
9. Lan, A.S., Baraniuk, R.G.: "A contextual bandits framework for personalized learning action selection. EDM (2016)
10. Sweeney, Mack, et al. (2016) "Next-term student performance prediction: A recommender systems approach." arXiv preprint [arXiv:1604.01840](https://arxiv.org/abs/1604.01840)
11. Chen, Yuying, et al. "Tracking knowledge proficiency of students with educational priors. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (2017)
12. Cen, H., Koedinger, K., Junker, B.: Learning factors analysis—a general method for cognitive model evaluation and improvement. In: International conference on intelligent tutoring systems. Springer, Heidelberg (2006)
13. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User-Adap. Inter.* **4**(4), 253–278 (1994)
14. Liu, S., et al.: A hierarchical memory network for knowledge tracing. *Expert Sys. Appl.* **177**, 114935 (2021)
15. Cai, Y., et al.: "Learning trend analysis and prediction based on knowledge tracing and regression analysis. In: International conference on database systems for advanced applications. Springer, Cham (2015)
16. Gan, W., Sun, Y., Sun, Yi.: "Knowledge interaction enhanced knowledge tracing for learner performance prediction. In: 2020 7th international conference on behavioural and social computing (BESC). IEEE (2020)
17. Yue, G., Beck, J.E., Heffernan, N.T.: "Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In: International conference on intelligent tutoring systems. Springer, Heidelberg (2010)
18. Pelánek, R.: Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Model. User-Adap. Inter.* **27**(3), 313–350 (2017)
19. Pardos, Z.A., Heffernan, N.T.: "Modeling individualization in a bayesian networks implementation of knowledge tracing. In: Priya, D. (ed.) International conference on user modeling, adaptation, and personalization. Springer, Berlin, Heidelberg (2010)



20. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: "Individualized bayesian knowledge tracing models. In: Priya, D. (ed.) International conference on artificial intelligence in education. Springer, Berlin, Heidelberg (2013)
21. Severinski, Cody, Ruslan Salakhutdinov.: "Bayesian Probabilistic matrix factorization: a user frequency analysis." arXiv preprint [arXiv:1407.7840](https://arxiv.org/abs/1407.7840) (2014).
22. Käser, T., et al.: "Dynamic bayesian networks for student modeling. *IEEE Transact Learn. Technol.* **10**, 450–462 (2017)
23. Piech, Chris, et al.: "Deep knowledge tracing." *Advances in neural information processing systems* **28** (2015)
24. Zhang, Jiani, et al. "Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th international conference on World Wide Web (2017)
25. Abdelrahman, Ghodai, and Qing Wang. "Knowledge tracing with sequential key-value memory networks." In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (2019)
26. Su, Yu, et al.: "Exercise-enhanced sequential modeling for student performance prediction." In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1. (2018)
27. Liu, Q., et al.: Ekt: Exercise-aware knowledge tracing for student performance prediction. *Transact. Knowl. Data Eng* **33**, 100–115 (2019)
28. Pandey, Shalini, and George Karypis.: "A self-attentive model for knowledge tracing." arXiv preprint [arXiv:1907.06837](https://arxiv.org/abs/1907.06837) (2019).
29. Choi, Youngduck, et al.: "Towards an appropriate query, key, and value computation for knowledge tracing." In: Proceedings of the Seventh ACM Conference on Learning@ Scale (2020)
30. Nakagawa, Hiromi, Yusuke Iwasawa, and Yutaka Matsuo. "Graph-based knowledge tracing: modeling student proficiency using graph neural network." In: 2019 IEEE/WIC/ACM International Conference On Web Intelligence (WI). IEEE (2019)
31. Song, X., et al.: Jkt: A joint graph convolutional network based deep knowledge tracing." *Inform. Sci.* **580**, 510–523 (2021)
32. Shen, Shuanghong, et al.: "Convolutional knowledge tracing: Modeling individualization in student learning process." In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (2020)
33. Huang, Q., et al.: NTM-based skill-aware knowledge tracing for conjunctive skills. *Comput. Intell. Neurosci.* **27**, 2022 (2022)
34. Chen, T., et al.: Xgboost: extreme gradient boosting." *R Package Vers. 0.4-2* **4**, 1–4 (2015)
35. Chen, Tianqi, and Carlos Guestrin.: "Xgboost: A scalable tree boosting system." In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (2016)
36. Torlay, L., et al.: Machine learning–XGBoost analysis of language networks to classify patients with epilepsy. *Brain Inf.* **4**(3), 159–169 (2017)
37. Stamper, J., et al.: Challenge data set from KDD Cup 2010 Educational Data Mining Challenge (2010)
38. Vie, Jill-Jênn, and Hisashi Kashima.: "Knowledge tracing machines: Factorization machines for knowledge tracing." In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33. No. 01 (2019)
39. Chen, T., et al.: Package 'xgboost.' *R Vers.* **90**, 1–66 (2019)
40. Mitchell, R., Frank, E.: Accelerating the XGBoost algorithm using GPU computing. *PeerJ Comp. Sci.* **3**, e127 (2017)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.