Using Multi-feature Embedding towards Accurate Knowledge Tracing

Yang Yu, Caidie Huang, Liangyu Chen, Mingsong Chen
MoE Engineering Research Center of Software/Hardware Co-Design Technology and Application
East China Normal University, Shanghai, China
lychen@sei.ecnu.edu.cn, mschen@sei.ecnu.edu.cn

***Abstract*—Knowledge tracing is a crucial task in intelligent tutoring systems. Aiming at the shortcomings of traditional knowledge tracing technology such as low prediction accuracy, overfitting and low utilization of multi-features, this paper proposes a knowledge tracing model SRGCA-M using multi-feature embedding with stacked residual GRU network. Compared with the traditional methods that only use the historical record of answering exercises, our approach utilizes a variety of features in the learning process of students to deep characterize students' learning. We increase the layers number of GRU network to expand the capacity of sequence learning and use residual connections to solve the problems of network degradation and vanishing gradient. We use the auto-encoder to solve the problem that the cross-feature encoding will rapidly increase the dimension of the input data. Comprehensive experimental results demonstrate that compared with various advanced techniques, our approach can not only achieve better performance of tracking knowledge changes of students but also fully utilize multi-feature information of students in the learning process.**

***Index Terms*—Knowledge Tracing; GRU; Multi-feature Embedding; Auto-Encoder**

I. Introduction

Intelligent tutoring systems (ITS) are computer-based educational systems that act as smart teachers to guide students' learning. Knowledge tracing is the key step in ITS to track the change process of the knowledge mastery of students according to the historical records of learning, predict their future learning performance, and better provide students with personalized learning guidance services. Some pedagogical researchers believe that the knowledge concepts investigated in the exercise may be specific and relevant, and the mastery of the knowledge investigated in the exercise will affect their performance in the exercise, which means that the exercise investigation is the manifestation of the cognitive state of students.

With the popularization of online education, a large number of exercise answering data of students have been generated on the Internet, including knowledge concept of exercise, students' answer scores, students' answer practice, students' answer times, etc. Enough data promote the research progress of knowledge tracing models. However, there are still some problems, such as inaccurate prediction results, slow convergence speed and low utilization of multi-features. These problems limit the application and promotion of knowledge tracing in education to a certain extent. Researches have shown that a variety of features are helpful for evaluating students and personalizing instruction. Therefore, the research on knowledge tracing can not only promote the development of knowledge tracing in the education industry, but also reduce the stress of teachers and improve learning efficiency of students. Therefore, the research on knowledge tracing is of great significance to intelligent education.

DOI reference number:10.18293/SEKE2022-142

In order to improve the prediction accuracy of the knowledge tracing model, this paper proposes a Stacked Residual Gate Recurrent Network using Multiple Features with Cross Encoding and Auto-encoder (SRGCA-M), which can use various data in the process of students' answering. First, we use the lightGBM algorithm to select the features with high importance, and the cross-feature encoding and one-hot encoding method to encode the selected features. Because cross-feature encoding will rapidly increase the dimension of input features, we utilize an auto-encoder to compress the input features, and the compressed features are input into the SRG for training and prediction. SRGCA-M is tested on the Riiid dataset and compared with lightGBM and DKT. In addition, three ablation experiments are executed to verify the effects of multi-feature encoding and auto-encoder compression. The results show that the SRGCA-M achieves the best performance. This paper makes the following three major contributions:

- To improve model performance, we make full use of multiple features in the learning process of students by utilizing lightGBM feature selection, multi-feature cross-encoding and auto-encoder. In contrast, traditional knowledge tracing methods utilize the historical answer records of exercises.
- We increase the layers of network to expand the capacity of sequence learning by using the stacking GRU network. Besides, the use of residual connections solve the problems of network degradation and gradient vanishing.
- We improve the performance of the model by using an auto-encoder to represent the features which addresses the problem that cross-feature encoding will rapidly increase the dimensionality of the input data.

The rest of this paper is organized as follows. Section 2 summarizes the related works on knowledge tracing. Section 3 demonstrates our SRGCA-M model. Section 4 presents the experimental results and discussion. Finally, Section 5

concludes the paper.

II. Related Works

In order to solve the problems of low accuracy of knowledge tracking and low utilization of multi-features, various knowledge tracking methods have been proposed. For example, Bayesian Knowledge Tracing (BKT) [1] is a typical model based on probability graph. The model uses a set of binary variables to model the knowledge space characteristics of students, and each variable represents whether students master some knowledge concepts. The knowledge tracing model based on probability graph uses pedagogical theory, which is highly interpretable. However, the prediction efficiency largely depends on the rationality of establishing the probability map. When the establishment of the probability map is not reasonable, the performance will be greatly reduced.

Knowledge Proficiency Tracing (KPT) model [2], based on Probabilistic Matrix Factorization (PMF) [3] is proposed for knowledge tracing task. This model can effectively improve the prediction performance using the expert-marked topic knowledge concept matrix (Q matrix). However, the matrix decomposition model cannot add relevant information other than the topic knowledge concepts, such as exercise discrimination and exercise difficulty.

The Deep Knowledge Tracing (DKT) model proposed by Piech et al. [4] is the basic model in the field of deep knowledge tracing, which is based on the Recurrent Neural Network (RNN). The prediction performance of DKT is better than the classical methods at that time [5]. However, the DKT model suffers from poor interpretability, long-term dependencies, and few learning features [6]. In order to solve these problems, many researchers are committed to in-depth research on DKT and put forward many new methods. Dong et al. [7] used Jaccard coefficient to calculate the attention weight between knowledge components in the model a-dkt, and combined LSTM and total attention value to get the final prediction result. Zhang et al. [8] used the method of feature engineering to add the dimension reduction of answer time, answer times and the first action to the input layer of LSTM by using an auto-encoder.

For many years, the knowledge tracing method based on RNNs [9] has been dominant for the following reasons: i) the method based on neural network does not need to select features manually; ii) online education platforms generate massive amounts of data. The score data of exercises is the most relevant explicit data in students' knowledge space, and it is also easy to obtain. Therefore, the neural network method based on score data is universal; iii) the calculation of RNN and its variants combines the output of past time information and integrates them into the calculation of current time. Therefore, in knowledge tracing, this kind of model can achieve good results on datasets with temporal information.

Therefore, in this paper, we propose a residual network based on GRU, which uses a stacked residual GRU network (SRG) to learn students' answer sequences, and uses residual connections to reduce the difficulty of model training.

III. Our Method

A. Problem Definition

The task of knowledge tracking is to track students' knowledge mastery level and predict their future performance according to the historical records of answering exercises. The input is represented by the following formula,

$$\begin{aligned} &D = \{S_1, S_2, \ldots, S_N\}, \\ &S_i = \{x_{i1}, x_{i2}, \ldots, x_{it}, \ldots, x_{iM}\}, \\ &x_{it} = \{q_{it}, a_{it}\}. \end{aligned} \tag{1}$$

Suppose that there are N students, each student answers M exercises and $S_i = \{x_{i1}, x_{i2}, \ldots, x_{it}, \ldots, x_{iM}\}$ makes a exercise sequence for the students, and D represents the student set. At time t, x_{it} contains two parts: i) the exercise that the student is answering at the current moment; ii) the learner's answer to the exercise a_{it}. When x_{it} is 0, it means that the student answered incorrectly, and 1 means that the student answered correctly. The student behavior sequence is encoded and input into the recurrent neural network for training, and the learner's knowledge mastery level is obtained through the prediction output layer. Finally, the correct rate of the student's answer at the next moment is predicted. The range is 0 to 1, indicating the prediction probability. There are two basic tasks of knowledge tracking: i) predict students' future answer performance, i.e., the correct answer rate in the next time step; and ii) track the changes of students' mastery of knowledge concept to facilitate personalized learning guide.

B. Overall Structure

The framework of SRGCA-M model is shown in Figure 1 and consists of the following parts: data preprocessing, feature selection, multi-feature encoding and deep learning prediction. In the data preprocessing stage, the original dataset needs to be cleaned and sorted to get a relatively complete dataset. In the feature selection stage, we use lightGBM algorithm to calculate the importance of features, obtain the feature importance ranking, and predict the students' scores as a comparative experiment. In the multi-feature encoding stage, cross-feature encoding and one-hot encoding are used to encode the selected important features and students' response features. The encoded features are compressed by auto-encoder (AE). Finally, in the deep learning prediction stage, the compressed feature is integrated into SRG model to track students' knowledge level and predict students' achievement.

C. Multi-Feature Selection

Saivastava et al. [10] pointed out that the performance of models using cross-features is improved compared to models using single feature. Inspired by this, we utilize lightGBM to process multi-feature data. Firstly, input the students' muti-feature answer data, use the histogram algorithm to find the feature with the maximum gain, and determine the optimal segmentation point of the decision tree according to the feature. Using leaf-wise leaf growth strategy with depth constraints to generate cart tree. Then calculate the residual

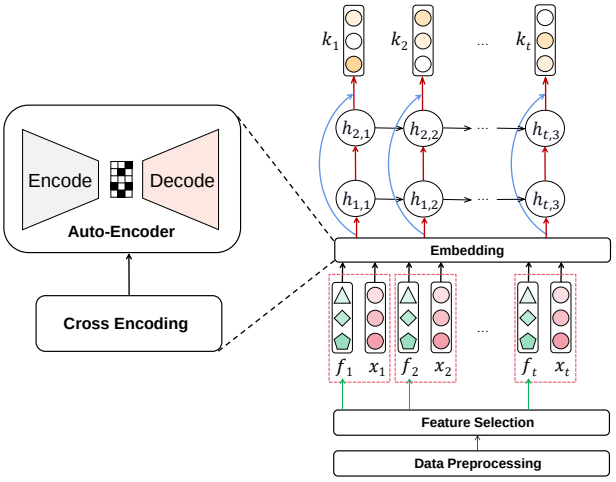

Fig. 1. SRGCA-M model framework.

value of cart tree, take the residual result of the previous tree as the training sample, train the next cart tree and repeat the training. Finally, the cart tree generated by each training round is weighted and summed to obtain the final prediction model.

LightGBM [11] algorithm measures the importance of feature attributes based on the number of times the feature is used as segmentation points. Sort the feature elements from large to small according to the attribute importance. Search from the complete set of all features, and judge whether to delete the feature with the lowest importance according to the result accuracy. Traverse all features and output the optimal feature subset. The input of the algorithm are the dataset D, the feature set $F = \{T_i \mid i = 1, 2, \ldots, d\}$, and the output is the optimal feature subset F_{best}.

D. Multi-Feature Encoding

After the features of the dataset are filtered by lightGBM algorithm, multiple features need to be encoded to form the input data. In this paper, we pose three multi-feature embedding methods: direct concatenating method, cross-feature encoding method and compressed cross-feature encoding method. Next, three embedding methods are introduced in detail.

Direct concatenating method forms a new vector by concatenating the answer data and the optimal feature directly. This method can simply convert a single feature vector into multiple feature vectors, which is the input x_t of the model.

Crossed features refer to the cross-encoding result of student answers and selected multi-features. The The cross-feature encoding method can be expressed by the following formula,

$$\begin{aligned} C\left(s_t, c_t\right) &= s_t + [\max(s) + 1] * c_t, \\ x_t &= O\left(C\left(s_t, c_t\right)\right) \oplus O\left(C\left(F_t, c_t\right)\right) \oplus O\left(F_t\right), \end{aligned} \tag{2}$$

where C represents cross encoding, O represents one-hot encoding, $\oplus$ sign indicates the concatenation of two vectors, i.e., $C = A \oplus B$, which indicates that vector B is spliced at the end of vector A. The number of rows of vector C is the same as that of A and B. The number of columns of vector C is the sum of that of A and B.), s_t represents the ID of the knowledge concept, and c_t represents the result of answer (1 means correct, 0 means wrong), F_t represents the optimal features selected by the LightGBM algorithm.

Because the cross-feature encoding will lead to the rapid increase of input dimension, the compressed cross-feature encoding method used in SRGCA-M utilizes an auto-encoder to compress the cross-encoded features. The specific calculation method is listed as follows,

$$\begin{aligned} z_t &= E\left(x_t\right) = \sigma\left(W x_t + b\right), \\ x_t' &= D\left(z_t\right) = \sigma\left(W x_t + b\right), \end{aligned} \tag{3}$$

where x_t is the input, the function E represents the encoding operation, and the function D represents the decoding operation. z_t represents the learned latent variable, which can be used as input data.

E. Stacked Residual GRU Network (SRG)

This paper uses the stacked residual GRU network to deal with the deep knowledge tracing task [12]. It improves the performance of traditional recurrent neural networks by increasing the number of network layers to expand the capacity of sequence learning. Besides, the residual connections help solve the problems of network degradation and gradient vanishing. The stacked residual GRU network SRG can be defined by the following formula,

$$\begin{aligned} h_{1,t} &= f_{gru-1}\left(h_{1,t-1}, x_t\right), \\ h_{2,t} &= f_{gru-2}\left(h_{2,t-1}, h_{1,t}\right), \\ k_t &= \sigma\left(W_{kt} h_{2,t} + b_k\right). \end{aligned} \tag{4}$$

The input x_t enters the first layer of GRU network to obtain the hidden variables $h_{1,t}$. Then the output of the first layer is used as the input of the second layer to obtain the output $h_{2,t}$ of the second layer GRU network. Then the knowledge level vector k_t is obtained from the full connection layer.

Because the increase of layers of the recurrent neural network will make it challenging to fit the model training, the SRG model introduces residual connection [13]. The addition of residual connections can make the training of stacked GRU network converge easily, so this paper proposes a stacked residual GRU network *SRG* with residual connections.

$$k_t = \sigma\left(W_{kh}\left(h_{2,t} \oplus x_t\right) + b_k\right). \tag{5}$$

Equation (5) reflects this idea by concatenating the input x_t of the model with the output $h_{2,t}$ of the hidden layer. Then input the concatenated vector to the next layer for prediction. The loss function is defined as follows:

$$\begin{aligned} L = &-\sum_t \left(a_t \log k_{t+1}\left(q_t\right) + \left(1 - a_t\right) \log\left(1 - k_{t+1}\left(q_t\right)\right)\right) \\ &-\sum_t \left(x_t \log x_t' + \left(1 - x_t\right) \log\left(1 - x_t'\right)\right). \end{aligned} \tag{6}$$

The whole loss is divided into two parts. The first part is SRG loss and the second part is the loss function of the auto-encoder, which also uses the cross-entropy loss function. x' is the reconstructed data generated from the encoder.

IV. EXPERIMENTS

To evaluate the effectiveness of our approach, we implement SRGCA-M based on Pytorch framework in this section. We evaluate and compare the effectiveness of SRGCA-M with other methods. All the experiments are conducted on a workstation computer with Centos 7 operating system, Intel i7-9700K CPU with 16 GB memory, and NVIDIA GRX2080Ti GPU with 11 GB memory.

Formally, we design substantial experiments to answer the following two research questions.

RQ_1**: (Effectiveness of multi-feature):** What is the performance of SRGCA-M using multiple features cross encoding and auto-encoder compared with SRG-S using single feature?

RQ_2**: (Effectiveness of cross encoding and auto-encoder):** What is the performance of SRGCA-M using multiple features cross encoding and auto-encoder compared with SRG-M using multiple features and SRGC-M using multiple features and cross encoding?

A. Dataset Configurations

The dataset used in the experiments is *Riiid*, which is derived from Riiid Answer Correctness Prediction, a student performance prediction competition on the Kaggle website. The Riiid dataset provides historical learning records of students, other students' performance on the exercises and other metadata of the exercises. All Riiid data are divided into three files: train.csv, questions.csv and collections.csv. The details of Riiid dataset are shown in Table I. In the experiments, the dataset is divided into training set and verification set according to the ratio of $4:1$.

TABLE I
STATISTICAL INFORMATION OF DATASET.

Dataset	Students	Knowledge Concepts	Records	Answers/Person
Riiid	174,954	187	41,667,551	238

Table II describes the specific field content of the questions.csv file. This file contains the relevant information of the question, such as id, correct answer and corresponding knowledge concepts. In the process of data preprocessing, it is necessary to compare the students' answer records in train.csv with the correct answers to the exercises in questions.csv to determine whether the students answered correctly.

TABLE II
QUESTIONS DATA CONTENT DESCRIPTION.

Field name	Field Description
question_id	ID of the problem
bundle_id	ID of the problem set
correct_answer	Right key
part	Relevant parts of TOEIC test
tags	One or more detailed label codes

The *train.csv* file contains multi-feature information about the exercises and the learning process of students. It includes whether the exercises are answered correctly, the time it takes to answer the exercise, the historical answering time, the time from the first interaction of students to the completion of the exercise, and the average time it takes to answer the previous set of exercises, whether students viewed explanations and correct answers after answering the previous set of exercises. Note that the students' learning is divided into two forms: watching lectures and answering exercises, and the information of watching lectures should be ignored in the records.

B. SRGCA-M Using Multiple Features

1) Experimental Settings: The baseline model in this experiment is lightGBM. At the same time, three groups of ablation experiments are set according to the characteristics, namely single feature SRG model (SRG-S), multi-feature SRG model (SRG-M) and multi-feature cross-encoding SRG model (SRGC-M).

LightGBM: lightGBM is the feature selection algorithm used in this experiment. LightGBM predicts students' scores through these feature training models. The results can get the importance ranking of features to the results. According to the feature importance ranking of lightGBM, the top-ranked features are selected for multi-feature encoding.

SRG-S: SRG-S is a single-feature SRG model. It uses a stacked residual GRU network to predict students' grades. The model is a single-feature model, which only predicts future grades through knowledge concepts and historical answer records of students.

SRG-M: SRG-M is a multi-feature SRG model based on SRG-S, where additional features are highly important features extracted from the lightGBM experimental results. SRG-M encodes students' answer records and additional features as the input of the prediction model. The model simply concatenates the features without cross-feature encoding or multi-feature compression.

SRGC-M: SRGC-M is a cross multi-feature SRG, which performs one-hot encoding and cross-feature encoding on student answer records and additional features, and takes the fused features as the input of the prediction model. The model adds cross-feature encoding based on SRG-M. There is no multi-feature compression like SRG-M.

SRGCA-M: SRGCA-M is a compressed cross-encoded multi-feature SRG proposed in this paper. The model compresses student answer records and additional features after one-hot encoding and cross encoding as the input of the prediction model. Based on the SRGC-M, an auto-encoder is used to compress the cross-encoded multi-features. Then the compressed latent variables are input into the prediction network SRG for training.

2) Parameter Settings: Parameters of LightGBM are set as follows. The number of leaves is 200. When building a weak learner, the proportion of random sampling of features is 0.75, the sampling frequency is set to 10, bagging fraction is set to 0.8, the maximum number of iterations is set to 10000. The early stop round is set to 10 which means stop the training if the 10 training times are not optimized, verbose evaluation is set to 50, which means information is output every 50 iterations. The learning rate of SRGCA-M is set to 0.001, the

maximum step size is set to 50, which means every 50 records are a set of input data, less than 50 data are filled with 0, the minimum batch number is set to 128.

We set up three ablation experiments to verify the effectiveness of the model, where each ablation model verifies the effectiveness of a corresponding module. The learning rate is set to 0.001, the maximum step size is set to 50, which means every 50 records are a group of input data, less than 50 data are filled with 0, the minimum batch number is set to 128. SRGCA-M model and its ablation experimental model are optimized by Adam optimizer. All models use the evaluation functions AUC, RMSE and F1 as performance evaluation metrics.

3) Feature Selection: We calculate new features from the original data after cleaning raw data and selecting four million answer records with relatively complete data. After the training of lightGBM algorithm, we get the importance ranking of these features, shown in figure 2. These features are ranked as follows: the interval time of students answering the exercises for the first time, the average correct rate of exercises, and the average correct rate of students, the interval time of students answering the exercises for the third time, average answering time of the exercises, average number of times to check the problem analysis, average answering time of students, time for students to answer the previous set of exercises, the interval time for students to look back after answering wrong exercises, the interval time of students answering the exercises for the second time, the average number of times students viewed the problem resolution and correct answers, the top category code of lecture, the number of correct answers, and whether students viewed the resolution and correct answers after answering the exercises. We remove the last three features with obvious low scores and select the remaining features which are useful for evaluating the knowledge level of students and predicting their future performance.

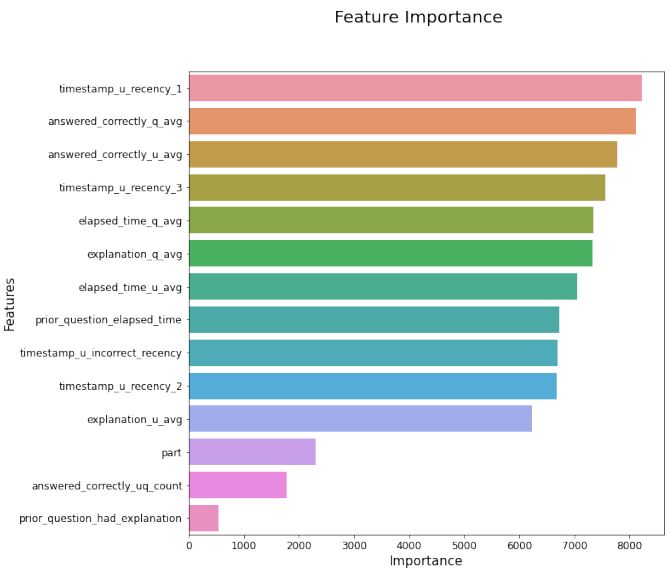

Fig. 2. Results of feature selection.

4) Student Achievement Prediction: The result of student achievement prediction is crucial in the performance evaluation of knowledge tracing task. In this experiment, lightGBM method is chosen as the comparative baseline, and the models SRG-S, SRG-M and SRGC-M are designed in the ablation experiment. The Riiid dataset is used in the experiment. We use the data after feature selection. The dataset is divided into training set and test set according to $4:1$. The *AUC*, *RMSE* and *F1* scores of each model on the Riiid dataset are recorded respectively.

Table III shows experimental results for different methods and Fig.3 compares them in visualization. One can easily observe that SRGCA-M achieves the highest AUC value and F1 score, while the RMSE value is also the lowest. Obviously, the prediction performance of SRG-based methods is much better than that of LightGBM. Except that the performance of SRGC-M using cross-feature encoding is lower than that of LightGBM, the performance of other SRG-based models is better than that of LightGBM. The performance of SRG-S using a single feature is similar to that of SRG-M, and slightly lower than the performance of SRGCA-M. In addition, The effect of SRG-M is worse than that of single-feature SRG-S. After adding additional features to SRG-M, the input dimension is increased by 11 times, so the training of the model will be more difficult. What's more, the performance of SRGC-M using cross-feature encoding is worse than the simply connected SRG-M. After adding additional features and crossed features to SRGC-M, the input dimension is increased by 22 times, so there are too many network parameters. Instead, the model becomes bloated and more difficult to converge. Importantly, after using the auto-encoder to compress the cross-multiple features, SRGCA-M has a great improvement than SRGC-M in the prediction performance compared , and the AUC value is improved by more than 16%.

TABLE III
TEST RESULTS OF EACH MODEL ON RIIID DATASET.

Model	AUC	RMSE	F1
LightGBM	0.778	0.423	0.771
SRG-S	0.866	0.298	0.831
SRG-M	0.818	0.411	0.795
SRGC-M	0.708	0.529	0.704
SRGCA-M	**0.868**	**0.296**	**0.833**

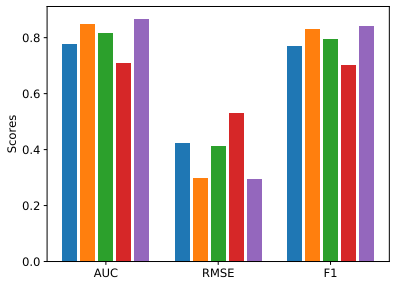

Fig. 3. Comparison of each model on Riiid Dataset.

In addition to predicting the correct rate of students answering questions at the next moment, another task of knowledge tracking is to track the change of students' knowledge level. Figure 4 is the visualization result of the data randomly selected from the Riiid validation set and predicted by the SRGCA-M model.

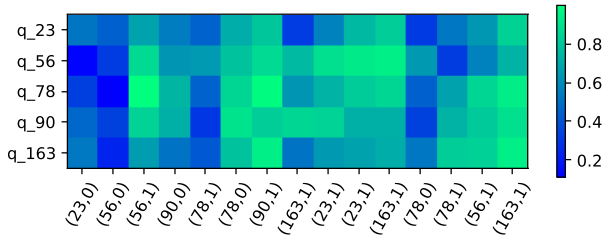

Fig. 4. Heatmap of SRG model tracking knowledge changes.

The horizontal axis represents the time series of a student answering questions. A two-tuple is used to represent the answer records. For example, the first record (23,0) represents the student's answer to question No. 23, and the answer is wrong. The vertical axis represents questions answered by students. The color of each square in the figure represents the student's mastery of knowledge at the current moment. The darker the color, the worse the student's mastery of the knowledge point corresponding to the question.

The color of each knowledge concept changes at different times, which indicates that the mastery level of knowledge concept is also changing accordingly. Focus on the first row, it is the changing process of students' mastery of knowledge concept No. 23. At the first moment, the student answered the question incorrectly, so the color of the corresponding square is dark. At the ninth and tenth moments, the student answered the question correctly, so the color of the square becomes light, and the color of the tenth moment is light. The change process of other knowledge concepts also has similar rules. From the results shown in Figure 4, we can sum up that the SRGCA-M model can effectively track the changes of mastery level of knowledge, which helps students to provide personalized tutoring services for learning guidance.

V. Conclusion

With the advent of the information age, the demand of people for online education is continuously increasing. As a key technology in intelligent tutoring systems, knowledge tracing has attracted many attentions. Although knowledge tracing technology has made great progress, there are still some problems. Aiming at the problems of inaccurate prediction results, slow convergence speed and low data utilization in knowledge tracing technology, this paper proposes a multi-feature knowledge tracing model SRGCA-M, which can effectively use the multi-feature information of students' learning history. SRGCA-M model first uses the lightGBM algorithm to filter student features for selecting the features with high importance to the results. The important features and historical answers of students are coded by cross-feature encoding method and one-hot encoding method. Because the feature dimension after encoding is too high, auto-encoder is used to compress the feature for better performance. Finally, the knowledge tracing model SRG is used to predict students' future performance and track students' knowledge mastery level. The experimental results show that the SRGCA-M model surpasses the lightGBM and DKT related models in prediction performance, and also gets better performance than other models in ablation experiments, which shows that our model can better track the knowledge level of students.

Acknowledgment

This work was supported by Natural Science Foundation of China 61872147 ,Shanghai Trusted Industry Internet Software Collaborative Innovation Center and Open Research Fund of Engineering Research Center of Software/Hardware Codesign Technology and Application, Ministry of Education (East China Normal University). Mingsong Chen and Liangyu Chen are the corresponding authors.

References

[1] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized bayesian knowledge tracing models," in *International Conference on Artificial Intelligence in Education (AIED)*. Springer, 2013, pp. 171–180.

[2] Y. Chen, Q. Liu, Z. Huang, L. Wu, E. Chen, R. Wu, Y. Su, and G. Hu, "Tracking knowledge proficiency of students with educational priors," in *Proceedings of Conference on Information and Knowledge Management (CIKM)*, 2017, pp. 989–998.

[3] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2008, pp. 1257–1264.

[4] C. Piech, J. Spencer, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," *ArXiv Preprint arXiv:1506.05908*, 2015.

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

[6] J. Lee and D.-Y. Yeung, "Knowledge query network for knowledge tracing: How knowledge interacts with skills," in *Proceedings of the International Conference on Learning Analytics & Knowledge (LAK)*, 2019, pp. 491–500.

[7] D. Liu, H. Dai, Y. Zhang, Q. Li, and C. Zhang, "Deep knowledge tracking based on attention mechanism for student performance prediction," in *Proceedings of International Conference on Computer Science and Educational Informatization (CSEI)*. IEEE, 2020, pp. 95–98.

[8] L. Zhang, X. Xiong, S. Zhao, A. Botelho, and N. T. Heffernan, "Incorporating rich features into deep knowledge tracing," in *Proceedings of ACM Conference on Learning @ Scale (L@S)*, 2017, pp. 169–172.

[9] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research (JMLR)*, vol. 15, no. 1, pp. 1929–1958, 2014.

[11] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 3146–3154, 2017.

[12] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (gru) neural networks," in *Proceedings of International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2017, pp. 1597–1600.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.