Kohama, H. et al. (Eds.) (2023). Proceedings of the 31st International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Recommending Learning Actions Using Neural Network

[a]*Hirokazu Kohama, Chubu University, Japan*
[b]*Yuki Ban, Chubu University, Japan*
[c]*Tsubasa Hirakawa, Chubu University, Japan*
[d]*Takayoshi Yamashita, Chubu University, Japan*
[e]*Hironobu Fujiyoshi, Chubu University, Japan*
[f]*Akitoshi Itai, Chubu University, Japan*
[g]*Hiroyasu Usami, Chubu University, Japan*

[a]tuna0724@mprg.cs.chubu.ac.jp
[b]banyuu@mprg.cs.chubu.ac.jp
[c]hirakawa@mprg.cs.chubu.ac.jp
[d]takayoshi@isc.chubu.ac.jp
[e]fujiyoshi@isc.chubu.ac.jp
[f]itai@isc.chubu.ac.jp
[g]usami@isc.chubu.ac.jp

Abstract: Many studies applying neural networks to the field of education have focused on student performance prediction and explainability of their decisions. While those studies introduced neural networks into educational settings, such networks cannot directly support student learnings in place of teachers. Therefore, we present a method that uses a general Transformer encoder to recommend appropriate learning actions for improving student performance. By considering the attention weight of a low-performing student to be close to that of a high-performing student, our method recommends the learning materials and actions for learning the materials. To evaluate the effectiveness of our method, we trained a deep neural network (DNN) on a private dataset of student operations (e.g., NEXT, PREV, OPEN) on digital learning materials obtained from a Japanese university. The number of operations divided by each learning material and by type of operation are input to the DNN, and the DNN outputs the student's grade on 5-point scale. We applied our method with this trained DNN to samples that successfully predicted grades, and the number of operations increased on the basis of the recommended learning materials and actions. By re-inputting modified sample into the DNN, we then observe how the student performance changes. The results of this simple experiment indicate that more students improved their performance with both the material-based and operation-based recommendations than with random recommendations. The percentage of students whose grades improved tended to be larger for those with low grades. Specifically, the improvement ratio for students with the two lowest grades was over 90% by operation-based recommendation. This is consistent with our intuition that low-performing students are more likely to improve.

Keywords: Student Performance Prediction, Explainable AI, Transformer

1. Introduction

Deep neural networks (DNNs) have been actively studied in various fields such as image processing and natural language processing. They are expected to be used for education in the current remote educational environment triggered by the COVID-19 pandemic (Adedoyin & Soykan, 2020). In an educational environment where there is physical distance between

teachers and students, it is difficult for teachers to provide individual feedback, so support from DNNs is necessary.

There have been many studies introducing DNNs into educational environments (Piech et al., 2015; Imran et al., 2019; Abdelrahman & Wang, 2019; Xing, & Du, 2019). These studies showed that we can use DNNs to predict a student's performance and possibility of early school dropout. However, it is necessary to incorporate explainability and accountability in educational DNNs design (Webb et al., 2021). There many studies explained the basis for DNN predictions (Baranyi et al., 2020; Mu et al., 2020; Vultureanu-Albişi & Bădică, 2021; Hasib et al., 2022; Swamy et al., 2022). However, these studies did not focus on DNNs replacing teachers, thus could not directly reduce the burden on teachers.

We propose a method with which artificial intelligence (AI) instead of teachers recommend appropriate learning actions (e.g., selection of learning materials, operations on learning materials) for improving student performance (see Section 2). We developed this method to misidentify the output of AI predicting student grades as a good grade by minimally perturbing the input of a student with a low grade. The pipeline for integrating our method into an educational environment is shown in Figure 1. We conducted an experiment to evaluate our method, and the results are presented in Section 3. Here, the predicted grades are denoted by S (excellent) , A (good), B (satisfactory), C (pass), and F (fail).

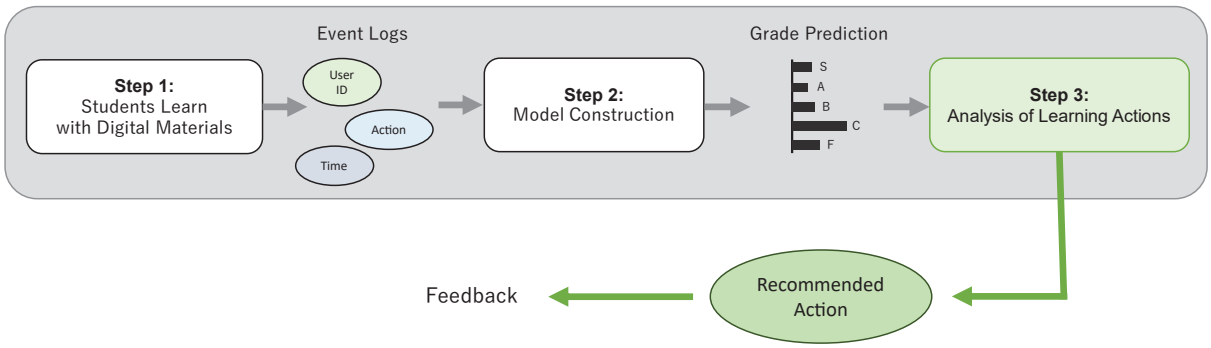

Figure 1. Pipeline for Deploying DNN into an Educational Environment.

2. Proposed Method

Given a DNN and dataset including student learning actions on digital-learning materials, we developed a method for recommending learning actions to improve grades in an input-data dependent manner. We first introduce the criterion to identify learning actions.

2.1 Identification of Learning Actions that Significantly Affect Predictions

When an input vector $\boldsymbol{x}$ is perturbed $\boldsymbol{\delta}$, the network loss $\mathcal{L}$ changes as:

$$\Delta\mathcal{L}(\boldsymbol{x}) = |\mathcal{L}(\boldsymbol{x}) - \mathcal{L}(\boldsymbol{\delta} \odot \boldsymbol{x})|, \tag{1}$$

where $\odot$ denotes the Hadamard product. Next, to compute the impact of each input, Equation (1) is transformed with first order Taylor expansion by focusing on a single element $x_i \in \boldsymbol{x}$:

$$\begin{aligned} \Delta\mathcal{L}(x_i) &= |\mathcal{L}(x_i) - \mathcal{L}(\delta x_i)| \\ &= \left|\mathcal{L}(x_i) - \mathcal{L}(x_i) - \frac{\partial\mathcal{L}}{\partial x_i}(\delta x_i - x_i) - \mathcal{O}(\|x_i\|^2)\right| \\ &\cong \left|\frac{\partial\mathcal{L}}{\partial x_i}(\delta x_i - x_i)\right|, \end{aligned} \tag{2}$$

where $\mathcal{O}$ denotes terms of higher order than the 2nd degree. We evaluate the input data on material or operation in accordance with Equation (2).

Materials: If the number of learning materials is M and the type of operations (e.g., NEXT, PREV, OPEN) is N, the input data are denoted as $\boldsymbol{X} \in \mathbb{R}^{M \times N}$. When $\boldsymbol{X}$ is input, the score of material m is expressed by using Equation (2) as the following equation:

$$S_m(\boldsymbol{X}) = \sum_{n=1}^{N} \left| \frac{\partial \mathcal{L}}{\partial x_{m,n}} \left(\delta x_{m,n} - x_{m,n} \right) \right|, \tag{3}$$

where $\boldsymbol{x}_{m,n}$ represents the number of operations per type of operation in m. We select the material with the highest score and recommend it to students.

Operations: The score for each operation is expressed as:

$$S_n(\boldsymbol{X}) = \sum_{m=1}^{M} \left| \frac{\partial \mathcal{L}}{\partial x_{m,n}} \left(\delta x_{m,n} - x_{m,n} \right) \right|. \tag{4}$$

We identify the type of operation that is effective for learning by using Equation (4). As with the material criterion, we recommend to students the operation with the highest score.

2.2 Optimization for Transformer

The theory introduced in Section 2.1 is applied independently of DNN structure. However, calculation of automatic differentiation using the loss between the correct label and output incurs large computational cost. Therefore, we introduce an additional lightweight criterion optimized for a Transformer (Vaswani et al., 2017).

A Transformer introduces an attention mechanism to determine the dependencies of various ranges (e.g., shorter, or longer range) within a sequence. In this attention mechanism, attention weight $\boldsymbol{A}$ in the first layer is derived using input $\boldsymbol{X} \in \mathbb{R}^{M \times N}$ as:

$$\boldsymbol{A} = \mathrm{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}} \right) \in \mathbb{R}^{M \times M}, \tag{5}$$
$$s.t. \boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}^q, \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}^k,$$

where d_k is a scaling coefficient, and $\boldsymbol{W} \in \mathbb{R}^{d \times M}$ denotes learnable parameters. By considering the interrelationships between input sequences using Equation (5), we can determine which inputs are contributing to the prediction.

Given an ideal attention weight $\boldsymbol{A}'$ of a high-performing student, we can reduce the computational cost from Equation (3) and (4). The method is simply to replace the loss in Equations (3) and (4) by the squared loss between $\boldsymbol{A}'$ and $\boldsymbol{A}$:

$$\mathcal{L} = \sum_{j=1}^{M} \sum_{i=1}^{M} \left(a'_{i,j} - a_{i,j} \right)^2, \tag{6}$$
$$s.t. a'_{i,j} \in \boldsymbol{A}', a_{i,j} \in \boldsymbol{A}.$$

This optimization method makes it possible to recommend learning actions only through processing by the first layer of Transformer encoders.

3. Experiment

We evaluated the recommendation of learning actions on a private dataset obtained in a real educational environment. To verify that the ideal attention weight is obtained from the training data, we first investigate the attention weight per grade. We then investigated the validity of recommending learning actions.

3.1 Dataset

We use log data of real student learning actions and the grades collected at a Japanese university. These log data were compiled for 21 types of operations (e.g., NEXT, PREV, OPEN) executed on 12 types of materials in a single course. The number of training data

was 114, and the number of test data was 51, because we used one year of such data for evaluation collected for a period of three years 2020, 2021 and 2022.

Figure 2 illustrates how the data were preprocessed. We treated the number of operations for each type of operation as a histogram for each learning material. Thus, the input for a student was $\boldsymbol{X} \in \mathbb{R}^{12\times 21}$. The values of these histograms were normalized by the maximum number of operations calculated using the histograms of all students.

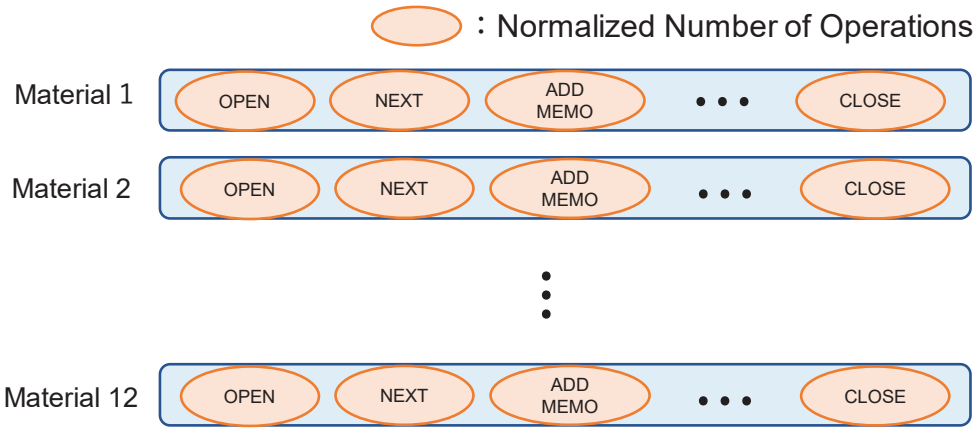

Figure 2. Preprocessing of The Data of One Student.

3.2 Experimental Setup

We construct a success-prediction model that can accurately classify the corrected label (final grade) y_s for student s from learning actions, as shown in Figure 2. We rely on a DNN architecture that is based on Transformer encoders, which can combine high with reduced computational cost due to Equation (6). An overview of the model structure is shown in Figure 3. The features input to the model are linear projections from student data such in Figure 2. The number of Transformer encoders L was 4. The model was trained using Adam (Kingma & Ba, 2015) for 200 epochs with an initial learning rate of 0.0001 and batch size of 32. The loss function uses cross-entropy loss.

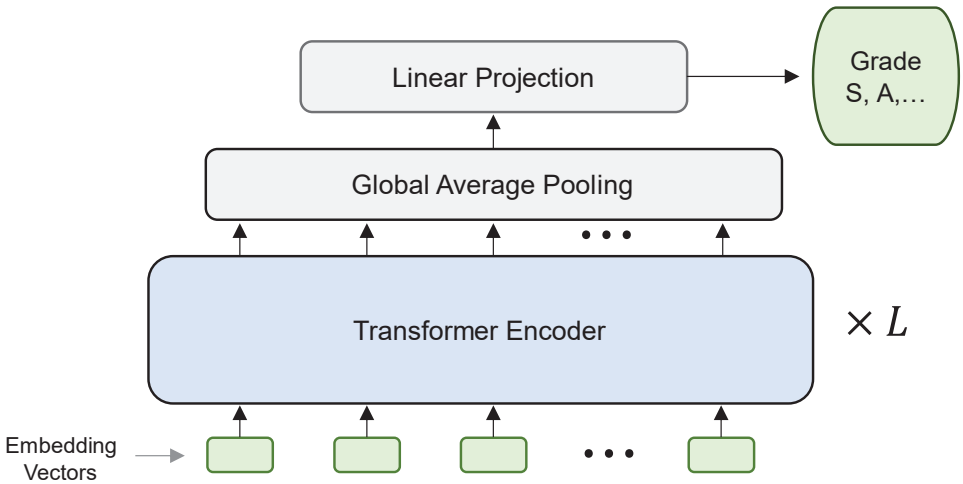

Figure 3. Overview of Model Structure for Grade Prediction.

3.3 Prediction of Student Grades

To coordinate the number of embedding dimensions $\boldsymbol{d}$ and heads $\boldsymbol{h}$, we compare

their prediction accuracy with Transformer models of various scales.

Table 1 shows the top-1 accuracy of grades on the test dataset. The number of embedding dimensions was investigated in the range from 64 to 256, and the highest test accuracy was achieved at 128. The highest accuracy was then obtained with one head in all comparisons. Using the results of this experiment as a guide, we recommend learning actions using a model with 128 embedding dimensions and one head.

3.4 Attention Weight Differences Between Grades

Table 1. *Test Accuracy of Grade Prediction*

d	h	Top-1 Accuracy [%]
64	1	39.21
64	2	39.21
128	1	**52.94**
128	2	47.06
256	1	45.10
256	2	45.10
256	4	27.12

Attention is conducted multiple times in parallel, which results in higher performance than just using a single head. However, our results were the exact opposite (see Section 3.3). This means that mixing features between materials does not require such complex patterns. We hypothesize that if the mixing between materials is represented by a simple pattern, there is an ideal attention weight, as described in Section 2.1, common to high-performing students. To confirm this hypothesis, we investigated the attention weight by grade.

Figure 4 shows the attention weights of the first layer, calculated using the training dataset. In Figure 4, high values of attention weight are shown in light colors and low values in dark colors. These attention weights are averaged per grade. Since the number of teaching materials was 12, the size of attention weight was 12x12. We observed that the attention weights of the high-performing students, such as those with grades S and A, had a biased attention to fewer materials. The attention of the lowest-performing students are plotted across many learning materials. Therefore, we use the attention weight of students with grade S as an ideal attention weight to reduce the computational cost of recommending learning actions.

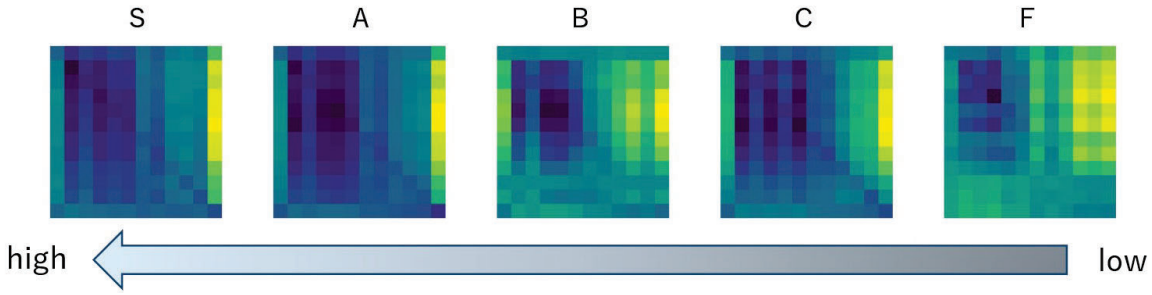

Figure 4. Average of Attention Weight per Grade Obtained from First Layer of Transformer Encoder.

3.5 Recommending Learning Actions

Our method recommends appropriate materials and operations to a student using the trained DNN.

Material Recommendation: Our method recommends the materials that students should be learning to improve their grades. To investigate such improvement, we increased the number of operations included in recommended materials by a factor of 1.5. We then re-input the sample with an increased number of operations into the DNN to check for changes in performance. Figure 5 shows the ratio of performance improvement for the sample that successfully predicted grade before modification when our method was used to recommend materials to be learned. For comparison, Figure 5 plots the improvement ratio with dashed lines when learning materials were randomly recommended (the results plotted with solid lines are from our method). The vertical axis is the ratio of predicted improved performance after re-input, and the horizontal axis is the number of materials that were modified using our method or randomly.

High rate of improvement was observed in the lowest-performing students with grades C and F. Their performance tended to improve more in proportion to the number of materials modified. These results are consistent with the intuition that lower-performing students are more likely to improve with a small amount of learning, and that students who learn more improve their performance. Then, the improvement ratio with our method is higher than that with random, which confirms the effectiveness of our method.

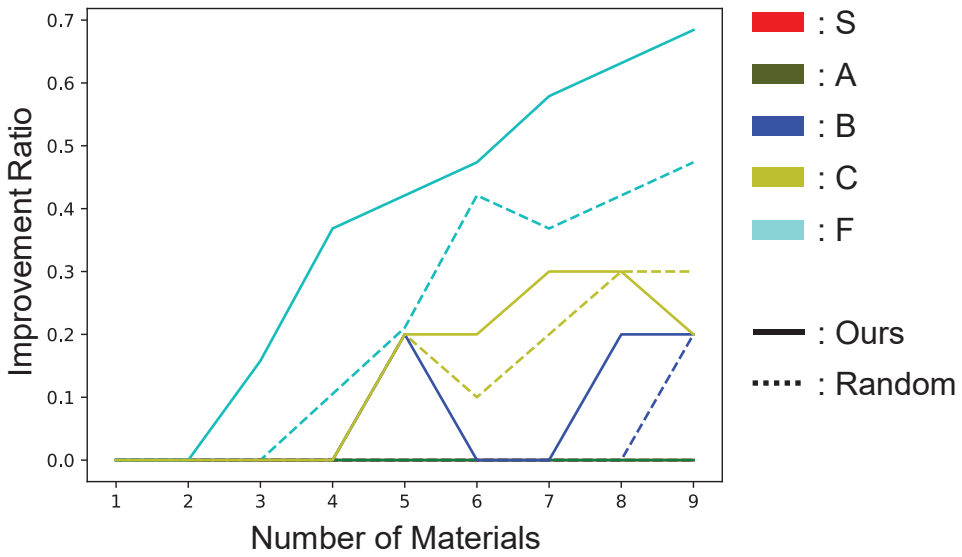

Figure 5. Ratio of Performance Improvement for Each Material Modified.

Operation Recommendation: We then identified the types of operations that should be used to improve student performance. By modifying the number of operations by a factor of 1.5 for our method, or for a randomly recommended operation, we can compare the changes in grade predictions. Figure 6 shows the ratio of performance improvement for the sample that successfully predicted grade before modification when our method was used to recommend operations to be learned. The vertical axis is the percentage of predicted grades that improved after

re-input, and the horizontal axis is the number of operations that were modified using our method or randomly.

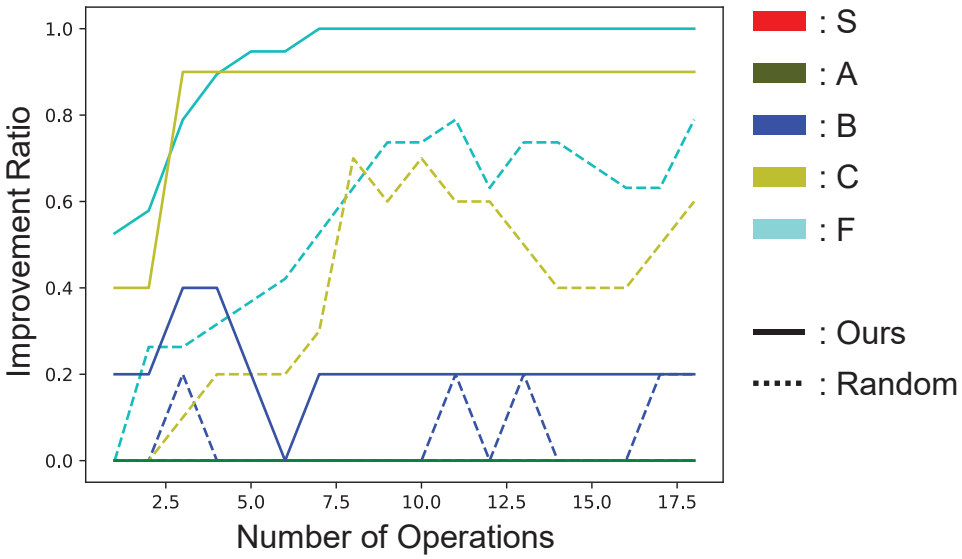

Figure 6. Ratio of Performance Improvement for Each Operation Modified.

The operations recommended with our method were more accurate than those randomly recommended. From these results, the performance of students with low grades improved significantly with our method.

4. Discussion

Our results showed that the suggested learning actions improved student performance. However, our method is highly dependent on the quality and quantity of the learning data and may not work well when the number of students is small or when there are not enough learning logs (our method may be very accurate in the presence of a large number of learning logs). Figure 7 shows the top-5 most frequently recommended operations and their average scores. Many basic operations (e.g., NEXT, PREV, OPEN) are recommended, and operations such as “ADD MEMO” are not included. While the basic operations are performed many times and are understandably effective for estimating student learning density, most realistic teachers would recommend leaving notes and bookmarking important passages. Training on a larger dataset is necessary to achieve the same performance as a teacher using a DNN. Therefore, a larger dataset would be needed to construct a DNN comparable with a teacher. Since suggesting learning actions using too small a dataset may invade the privacy of certain highest-performing students, one should be careful when deploying a pipeline such as that shown in Figure 1.

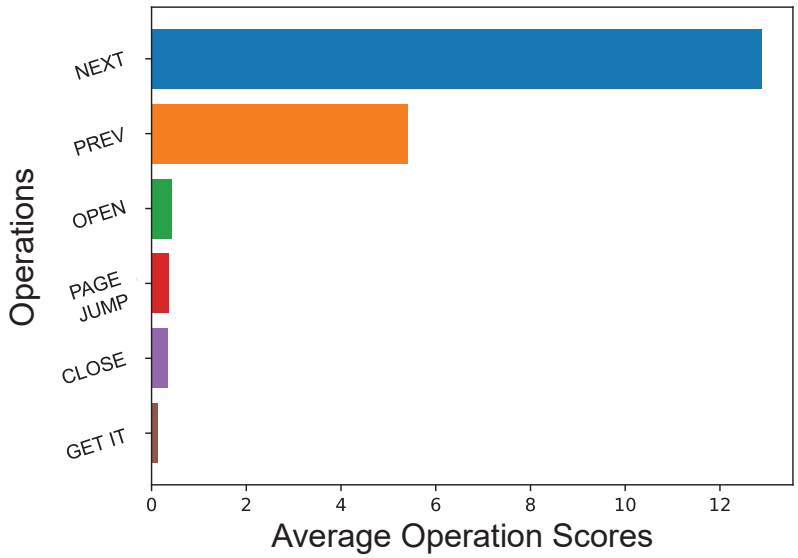

Figure 7. Top-5 Recommended Operations and Their Scores.

5. Conclusion

We showed that considering the impact of inputs on grade prediction can recommend the learning actions needed to improve grades. Learning actions are recommended on a per-material or per-operation basis, allowing the DNN to provide detailed feedback to the students. In particular, the operation recommendations confirmed a higher improved ratio of grades than the learning material recommendations. Recommending learning actions to students is a necessary component for neural networks to directly assist teachers and encourage the introduction of AI into the educational environment. In future work, we plan to use data augmentations suitable for event logs to recommend learning actions that are more in line with our intuition.

Acknowledgements

This work was supported by JST CREST Grant Number JPMJCR22D1, Japan.

References

A. S. Imran, F. Dalipi, & Z. Kastrati. (2019). *Predicting student dropout in a MOOC: An evaluation of a deep neural network model.* 5th International Conference on Computing and Artificial Intelligence.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, & I. Polosukhin. (2017). *Attention Is All You Need.* Advances in Neural Information Processing Systems.

A. Vultureanu-Albişi & C. Bădică. (2021). *Improving students' performance by interpretable explanations using ensemble tree-based approaches.* IEEE 15th International Symposium on Applied Computational Intelligence and Informatics.

C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, & J. Sohl-Dickstein. (2015). *Deep Knowledge Tracing.* Advances in Neural Information Processing Systems.

D. P. Kingma & J. Ba. (2015). *Adam: A Method for Stochastic Optimization.* 3rd International Conference on Learning Representations.

G. Abdelrahman & Q. Wang. (2019). *Knowledge tracing with sequential key-value memory networks.* 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.
K. M. Hasib, F. Rahman, R. Hasnat, & M. G. R. Alam. (2022). *A machine learning and explainable AI approach for predicting secondary school student performance.* IEEE 12th Annual Computing and Communication Workshop and Conference.
M. Baranyi, M. Nagy, & R. Molontay. (2020). *Interpretable deep learning for university dropout prediction.* 21st Annual Conference on Information Technology Education.
M. E. Webb, A. Fluck, J. Magenheim, J. Malyn-Smith, J. Waters, M. Deschênes, & J. Zagami. (2021). *Machine learning for human learners: opportunities, issues, tensions and threats.* Educational Technology Research and Development.
O. B. Adedoyin & E. Soykan. (2020). *Covid-19 pandemic andonlinelearning: the challenges and opportunities.* InteractiveLearning Environments (2020).
T. Mu, A. Jetten, & E. Brunskill. (2020). *Towards suggesting actionable interventions for wheel-spinning students.* International Educational Data Mining Society.
V. Swamy, B. Radmehr, N. Krco, M. Marras, & T. Käser. (2022). *Evaluating the Explainers: Black-Box Explainable Machine Learning for Student Success Prediction in MOOCs.* International Educational Data Mining Society.
W. Xing & D. Du. (2019). *Dropout prediction in MOOCs: Using deep learning for personalized intervention.* Journal of Educational Computing Research.