

Design Reflective Practice Assessment Tools for Teacher on Learning Management System Using AI Adaptive Feedback

Ikhsan, Universitas Terbuka, Indonesia

Muhammad Ashar, State University of Malang, Indonesia

Akhmad Khanafi Mashudi, State University of Malang, Indonesia

Ichwan, Universitas Terbuka, Indonesia

Kadarisman, Universitas Terbuka, Indonesia

Muhammad Iqbal Akbar, State University of Malang, Indonesia

The Asian Conference on Education & International Development 2024
Official Conference Proceedings

Abstract

This research introduces an innovative framework for enhancing teacher professional development at ICE Institute and open university through the integration of artificial intelligence (AI) into reflective practice assessment tools within the Learning Management System (LMS). The primary goal is to offer educators personalized and data-driven insights to refine their instructional practices and foster continuous improvement. The proposed reflective assessment tool employs AI algorithms to systematically collect and analyse data, including student performance, engagement metrics, and interaction patterns specific to the open and distance education context. The AI-driven adaptive feedback system provides real-time, personalized feedback, emphasizing strengths, pinpointing areas for improvement, and suggesting targeted instructional strategies. Seamlessly integrated into the ICE Institute LMS, the user-friendly interface features a visual analytics dashboard, benchmarking capabilities, and direct access to professional development resources. The tool encourages teachers to formulate individualized continuous improvement plans based on the feedback, ensuring a tailored approach to professional growth. Privacy and ethical considerations are paramount, aligning the tool with ICE Institute commitment to data security and ethical AI use. The scalability of the tool facilitates widespread adoption, fostering a collaborative community of educators dedicated to refining teaching practices within the unique context of open and distance education. This research represents a significant advancement in leveraging AI to elevate reflective practices among teachers at ICE Institute, ultimately enhancing the quality of education delivery in the digital learning landscape.

Keywords: Assessment-Tools, LMS, AI-Driven Adaptive Feedback, Digital-Learning

iafor

The International Academic Forum

www.iafor.org

Introduction

In the era of information overload, recommender systems have become an essential tool in various domains, including Learning Management Systems (LMS). Recommender systems are widely used to help users find their desired items or services from a large collection of options. They can improve user satisfaction, loyalty, and retention, as well as generate revenue for the providers. However, designing effective recommender systems is challenging, as they need to deal with complex and dynamic user preferences, item features, and system environments.

The recommendation system plays a crucial role in enhancing the user experience in the Learning Management System (LMS). According to, although LMS has facilitated access to various learning materials, the primary challenge is how to create a personalized and efficient learning experience for each user. They also state that many current LMS still function primarily as information repositories. The recommendation system in LMS can provide recommendations for learning materials that align with individual interests, abilities, and needs by analyzing user learning patterns.

One promising approach to address these challenges is to use deep reinforcement learning (DRL), which combines deep neural networks with reinforcement learning (RL). Recently, Deep Reinforcement Learning (DRL) has shown promising results in various domains, including recommendation systems [2]. DRL can model the sequential interactions between users and items, and optimize the long-term user engagement, which is crucial for LMS.

In recent years, there has been a surge of interest in applying DRL to recommender systems, and many novel methods and applications have been proposed. Therefore, in this article, we aim to provide a timely and thorough review of the state-of-the-art DRL methods for recommender systems and discuss the challenges and opportunities for future research. This article proposes an innovative approach to implementing a DRL-based recommender system in an LMS, leveraging the DRL framework from "A Deep Reinforcement Learning Based Long-Term Recommender System" [3]. This framework has demonstrated impressive results in optimizing long-term recommendation accuracy, making it an ideal choice for our LMS recommender system.

We hope that this work will not only contribute to the ongoing research in the field of DRL and its application in recommender systems but also inspire further innovation in the application of DRL in LMS recommender systems.

Related Works

In the context of recommender systems, a significant contribution has been made by [3] of a study on a deep reinforcement learning-based long-term recommender system. This study proposed an innovative top-N interactive recommender system that leverages deep reinforcement learning to optimize long-term recommendation accuracy and adapt to user preferences over time. The recommendation process was modeled as a Markov decision process, where the agent (recommender system) interacts with the environment (user) and learns from the feedback. The agent uses a recurrent neural network to generate the recommendation list and a policy gradient algorithm to update the parameters. The model was evaluated on three real-world datasets and compared with several baselines. The results demonstrated that the model outperformed the baselines in terms of hit rate and NDCG for the long-term recommendation and could handle both cold-start and warm-start scenarios. The

study made four main contributions and discussed the implications of the proposed model for future research and applications.

Overall Framework

The main framework used in this research is based on the Deep Reinforcement Learning Based Long-Term Recommender System framework, particularly in the use of the pre-built warm-start model. The recommender system operates as a Markov Decision Process, where the agent interacts with the environment sequentially. The model learning process will be divided into two stages: supervised learning and reinforcement learning.

A. RNN

The input layer is designed to manage sequential data between users and courses over time. In addition to containing temporal information, the input data also includes labels representing the final values corresponding to the courses taken by the user. These values are then fed into the input layer. The RNN layer utilizes EMGRU, proven to capture more complex information compared to GRU. The output layer generates recommendations in the form of a Top-N list, using the softmax activation function that converts raw outputs into probability distributions for each course.

B. Supervised Learning

Supervised learning pre-training involves using historical user interaction data, which includes the courses taken and their corresponding final grades. In supervised learning, two hierarchical target variables are employed: the selected course and the final grade associated with that course.

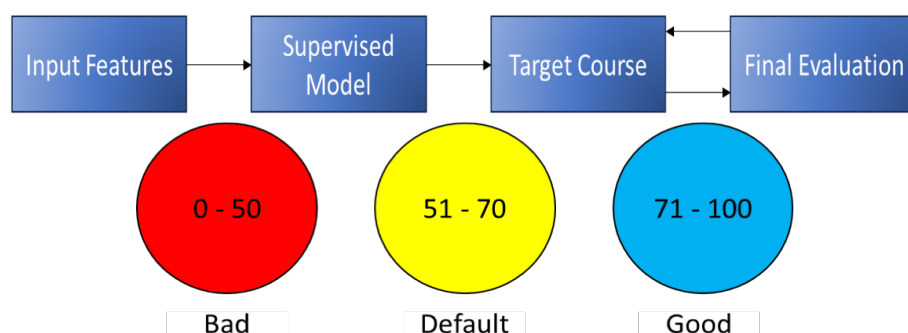


Fig. 1. Hierarki of historical user interaction data based supervised learning.

We apply categorization for value intervals, with different treatments for each target course value within a specific interval. Here, bad is equivalent to a negative value, while default and good are equivalent to positive values. Due to the nature of course values, which cannot be randomly assigned during the training process, we implement the creation of dummy data based on similar information from different users.

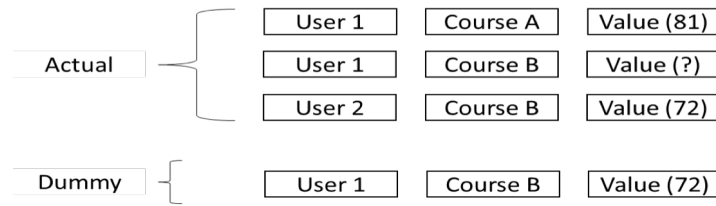


Fig. 2. Pengkategorisasian data process

Meanwhile, for user value data that has no similarity with other data, we apply the default value. This data is used to train the EMGRU model, which focuses on predicting the next Top-N recommended courses. The pre-trained weights saved by the model will serve as the foundation for the subsequent Reinforcement Learning (RL) training, allowing the RL agent to start training with knowledge of user preferences and sequential dynamics. This approach aims to build an initial representation of states and course relationships that are robust and provide an effective starting point for RL training.

C. Reinforcement Learning

Reinforcement Learning (RL) involves transitioning from a Supervised Learning (SL) environment to an RL environment aimed at optimizing policies for long-term recommendations while remaining adaptable to dynamic user preferences [5]. The process begins with initializing the agent with pre-trained weights obtained from the SL model using EMGRU. The agent then interacts with the environment or user, receives a state, takes action or recommends a course, and receives rewards based on user feedback. Rewards are not only based on the courses taken by the user but also on the final grades obtained in those courses. For each grade category (bad, default, good), different rewards are assigned, and the policy is updated using the RL algorithm called REINFORCE, which utilizes a policy gradient approach to maximize cumulative rewards. This learning process enables the agent to adapt to changing user preferences and dynamic real-world contexts, enhancing the model's effectiveness in generating accurate personalized recommendations.

Algorithm 1: Supervised Learning

Require:

- Training set U (user interaction data)
 - Learning rate η
 - Maximum sequence length B
 - EMGRU model architecture
- 1: while not stop do
 - 2: for each sequence (u, I_u) in U do
 - 3: Initialize: - count $\leftarrow 0$ - $s_u, 0 \leftarrow \text{zeros}$ (initial state representation)
 - 4: while count $< |I_u|$ do
 - 5: Generate training batch: - batch_sequences, batch_targets $\leftarrow \text{GetNextBatch}(I_u, \text{count}, B)$
 - 6: Forward pass: - predictions $\leftarrow \text{EMGRU}(\text{batch_sequences})$
 - 7: Compute loss: - loss $\leftarrow \text{LossFunction}(\text{predictions}, \text{batch_targets})$
 - 8: Backward pass and optimization: - $\theta \leftarrow \theta - \eta * \nabla_{\theta} \text{loss}$
 - 9: count $\leftarrow \text{count} + B$
 - 10: end while
 - 11: end for
 - 12: end while

Algorithm 2: Reinforcement Learning*Require:*

- *Trained EMGRU model from SL stage*
 - *Training set U (user interaction data)*
 - *Learning rate η*
 - *Maximum episode length B*
 - *Discount factor γ*
- 1: *while not stop do*
 - 2: *for each user u in U do*
 - 3: *Initialize: - count $\leftarrow 0$ - $su,0 \leftarrow$ zeros (initial state representation) - $mu \leftarrow$ ones (mask for uninteracted items)*
 - 4: *while count $< |I_u|$ do*
 - 5: *Generate episode: - $E, mu \leftarrow \text{GenerateEpisode}(I_u, mu, su,0, B)$ - $[su,1, \hat{a}_u,1, f_u,1, V_u,1, \dots, su,B, \hat{a}_u,B, f_u,B, V_u,B] = E$*
 - 6: *Calculate rewards: - $R_{u,t} \leftarrow \text{CalculateRewards}(E)$*
 - 7: *Update model parameters: - $\theta \leftarrow \theta + \eta * \sum_{t=1}^B \gamma^{(t-1)} * R_{u,t} * \nabla_{\theta} \log \pi(\hat{a}_t | su,t)$*
 - 8: *Update state and mask: - count $\leftarrow$ count + B - $su,0 \leftarrow su,B$ - $mu[i] \leftarrow 0$ for each item i in E (exclude interacted items)*
 - 9: *end while*
 - 10: *end for*
 - 11: *end while*

*Note:**U: Training set containing user interaction data.* *η : Learning rate, controlling the model's update speed during training.**B: Maximum sequence length, the number of items considered in a single training batch.* *θ : Model parameters (weights and biases) to be learned during training.**count: Counter tracking progress within a sequence.* *$su,0$: Initial state representation for a given sequence.* *∇_{θ} : Gradient operator, representing partial derivatives of a function with respect to model parameters θ .* *$\sum_{t=1}^B$: Summation notation, summing values from $t=1$ to B .* *$\gamma^{(t-1)}$: Discount factor raised to the power of $(t-1)$, used for discounting future rewards in RL.* *(u, I_u) : A tuple representing a user u and their interaction history I_u .* *$|I_u|$: Length of the interaction history for user u .***Design System**

Designing reflective practice assessment tools for teachers on a Learning Management System (LMS) with AI adaptive feedback involves creating a system that allows teachers to reflect on their teaching practices and receive personalized feedback based on AI analysis. Here's a flow model system for such a tool:

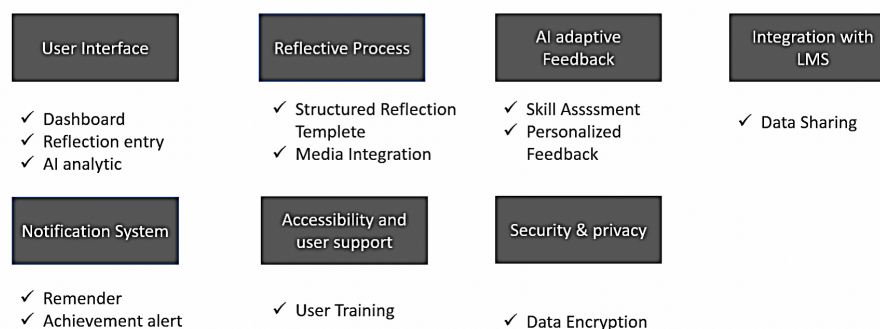


Fig. 3. Tools Design AI Learning Management System

We focus of the system development to be implemented in this AI-based LMS service is: Skill Assessment: Automated evaluation by AI of teaching skills and mapping strengths and areas for development.

Personalized Feedback: Tailored feedback with professional development recommendations. Suggestions for additional learning resources.

Assessment Tools: The LMS is designed for the use of Artificial Intelligence (AI) in assessment tools to enhance efficiency, objectivity, and provide in-depth insights into the performance or understanding of individuals (learners). In detail, the system design for assessments is created with automated evaluation, which can be in the form of Multiple Choice Exams, where the AI system can automate the assessment of multiple-choice answers without human intervention.

Furthermore, in enhancing the performance of the LMS with AI implemented in assessments, a design is also created for the evaluation system of Short and Long Answers, which will utilize Interactive Simulations. This involves the creation of interactive simulations overseen by AI to assess collaboration, leadership, or problem-solving skills.

Soft Skills Assessment: The implementation of the design in this system utilizes AI to analyze participant performance, adjusting the difficulty level and next materials according to individual needs. The final stage of the system's performance is the Learning Recommendations, tasked with providing learning recommendations based on assessment results to improve areas that require special attention.

Results and Discussion

Selection of the recommendation algorithm in the AI-based LMS involves a hybrid model called the Deep Reinforcement Learning-Based Long-Term Recommender System. This model integrates the assessment of learning outcomes as input for the recommendation system and incorporates personalized elements, such as adjusting the learning pace or suggesting additional course materials based on individual needs of the learners.

In the end, incorporating an interactive feedback mechanism allows users to provide feedback on the recommendations given. It is crucial to continually monitor and evaluate the performance of the recommendation system, integrating user feedback and learning data to enhance the accuracy and relevance of the recommendations.

The following are the results of the AI LMS design on the interactive online course platform:

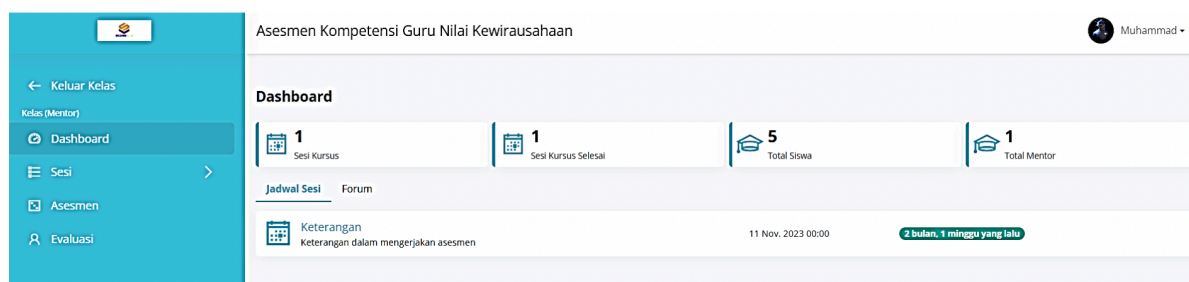


Fig. 4. Visualization of points regarding assessment results, including average scores, score distribution, and trends over time.

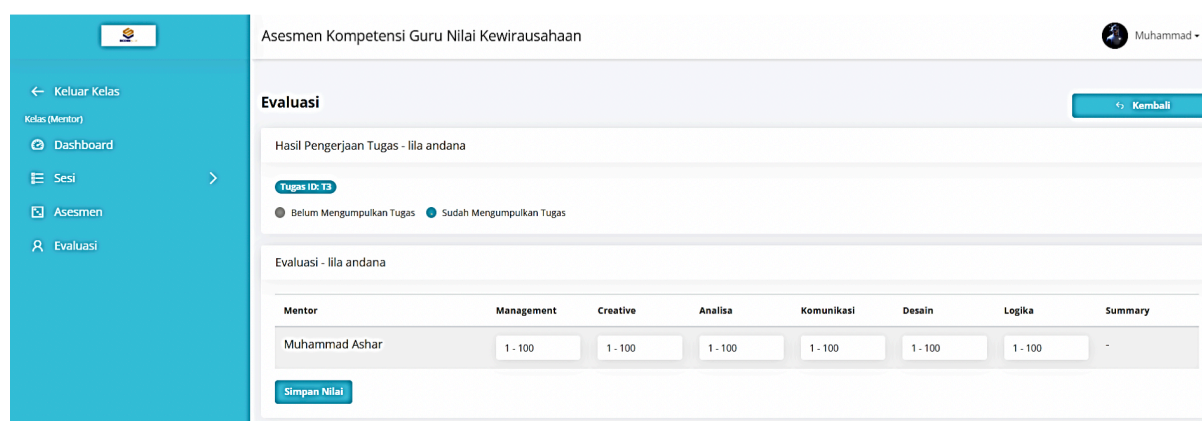


Fig. 5. Points to indicate the overall sentiment in student responses or reflections.

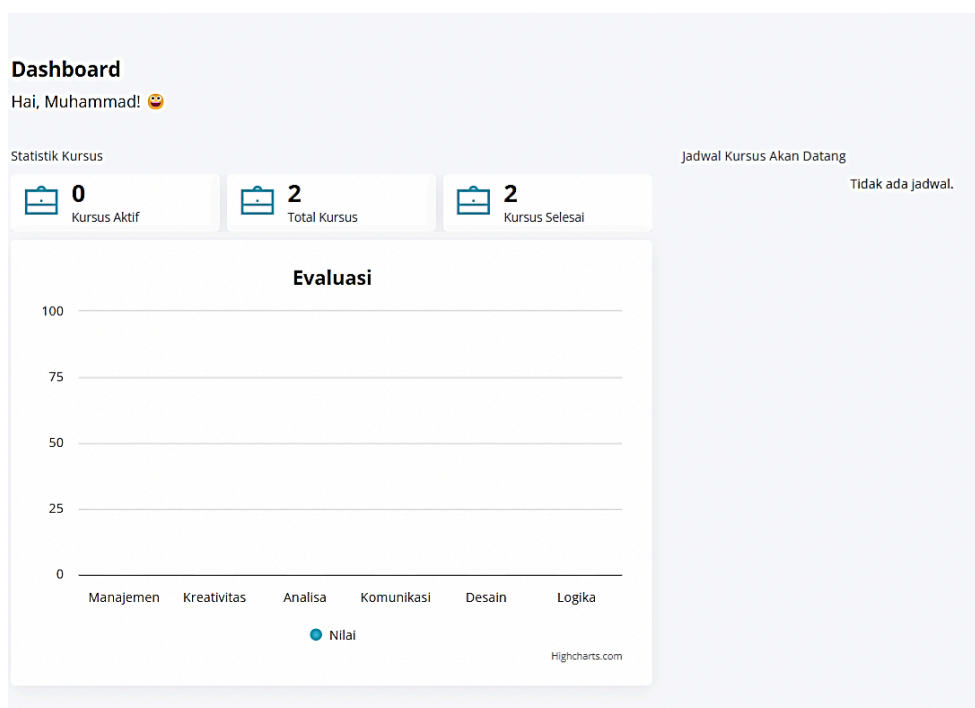


Fig. 6. Line chart depicting the progress of assessment results over time. Detailed captions of response outcomes to understand specific errors or strengths.

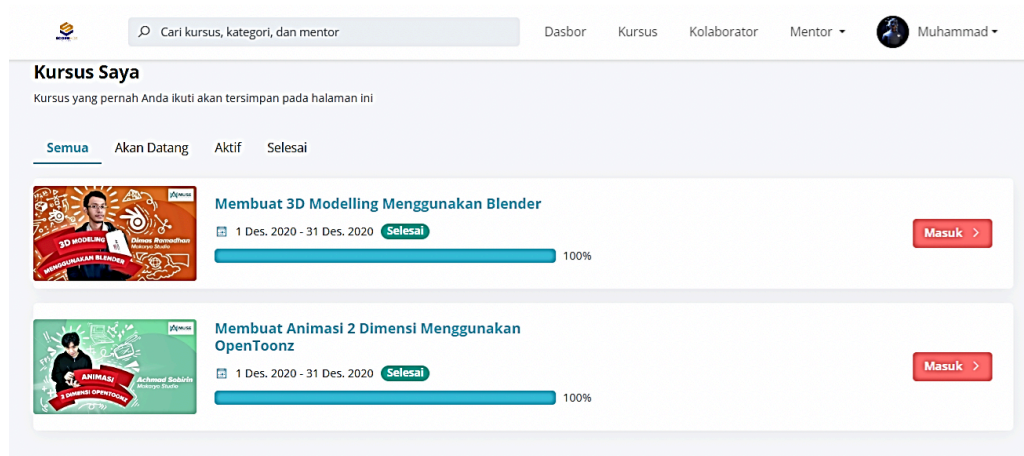


Fig. 7. List of learning or course recommendations based on assessment results and individual learning needs.

Conclusion

The research, therefore, presents an innovative approach towards building a personalized and adaptive AI feedback system. By incorporating final evaluations into the recommendation process, the model aims to provide more supportive and tailored recommendations, reflecting a comprehensive understanding of user preferences and performance across various courses. This adaptive feedback mechanism contributes to the overarching goal of creating an intelligent system capable of generating nuanced recommendations that go beyond traditional course data, thereby enhancing user experience and support. Finally, Learning recommendations assigned to provide personalized learning recommendations based on assessment results to address areas that require special attention.

References

- Afsar, M. M., Afsar, T. Crump, and B. H. Far. (2021). Reinforcement Learning Based Recommender Systems: A Survey. *CoRR*, vol. abs/2101.06286. [Online]. Available: <https://arxiv.org/abs/2101.06286>
- Chen, X. L., Yao, J. J. McAuley, G. Zhou, and X. Wang. (2021). A Survey of Deep Reinforcement Learning in Recommender Systems: A Systematic Review and Future Directions. *CoRR*, vol. abs/2109.03540, 2021, [Online]. Available: <https://arxiv.org/abs/2109.03540>
- Chung, J., C. Gulcehre, K. Cho, and Y. Bengio. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.
- Fan, C., M. Chen, X. Wang, J. Wang, and B. Huang. (2021). A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data. *Front Energy Res*, vol. 9, 2021, doi:10.3389/fenrg.2021.652801
- Huang, L., M. Fu, F. Li, H. Qu, Y. Liu, and W. Chen. (2021). A Deep Reinforcement Learning Based Long-Term Recommender System. *Knowledge Based System*, vol. 213. 106706, 2021, doi:<https://doi.org/10.1016/j.knosys.2020.106706>
- Murad, D. F., Y. Heryadi, B. D. Wijanarko, S. M. Isa, and W. Budiharto. (2018). Recommendation System for Smart LMS Using Machine Learning: A Literature Review, in *2018 International Conference on Computing, Engineering, and Design (ICCED)*, 113–118.
- Sutton R. S., A. G. Barto. (2019). *Reinforcement Learning: An Introduction*. MIT Press.
- Zhang, J.B. Hao, B. Chen, C. Li, H. Chen, and J. Sun. (2019). Hierarchical Reinforcement Learning for Course Recommendation in MOOCs. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 435–442.

Contact emails: ikhsan@ecampus.ut.ac.id
 muhammad.ashar.ft@um.ac.id
 ichwan@ecampus.ut.ac.id
 kadarisman@ecampus.ut.ac.id
 iqbal.akbar.ft@um.ac.id