An Approach for Dealing with Sequential Data in Intelligent Tutoring Systems

Nguyen Xuan Ha Giang[1,2], Lam Thanh-Toan[2], Nguyen Thai-Nghe[1*]
[1]Cantho University, Cantho City, Vietnam
[2]Cantho University of Technology, Cantho City, Vietnam
Correspondence: Nguyen Thai-Nghe. Email: ntnghe@cit.ctu.edu.vn
Communication: received 23/02/2025, revised 24/06/2025, accepted 04/07/2025
Digital Object Identifier: 10.32913/mic-ict-research.v2025.n2.1355

Abstract: The use of educational data to gain deeper insights into learners' interaction histories with Intelligent Tutoring Systems (ITS) is receiving increasing attention, especially in the context of online learning and the growing demand for digital transformation in education. Predicting learners' academic performance through the analysis and evaluation of their recorded activities in ITS plays a critical role in supporting educational administrators and instructors. An understanding of learners' abilities helps refine teaching methods and optimize learning environments, ultimately enhancing educational quality. Our research improves upon our previous work, which utilized only LSTM, by incorporating an ensemble model - CL-PSP, combining CNN and LSTM networks. Specifically, the study focuses on predicting learners' CFA capability - the likelihood of learners answering correctly on their first attempt. Knowledge evolves over time, observed in users' interaction preferences within session-based recommendation systems. CL-PSP leverages critical factor in shaping learners' academic performance in two educational datasets, KDD Cup 2010 and Assistment 2017. Several enhancements, including a revised ensemble architecture, improved error measurement, and refinements in data preprocessing. Results demonstrate that the proposed model significantly outperforms existing models, achieving superior performance with a lower Root Mean Square Error (RMSE). On the KDD Cup 2010 dataset, the model achieves a minimum RMSE of 0.375, while notable improvements are also observed on the Assistment 2017 dataset, further underscoring the model's effectiveness and robustness. The experimental results underscore the feasibility and considerable potential of utilizing session-based data in ITS to enhance both learning performance and educational quality.

Keywords: *Ensemble CNN-LSTM, Intelligent Tutoring Systems; Performance prediction; Session-based recommender system.*

I. INTRODUCTION

Intelligent Tutoring Systems are computer-based teaching and learning management systems that provide personalized assessments of individual competencies [1]. Within their operational architecture, ITSs enhance interactive environments, learning experiences for both learners and instructors through the integration of artificial intelligence techniques, including: *(1) Machine learning techniques*: encompass methods such as classification, clustering, regression, and reinforcement learning, which are utilized to predict learners' abilities based on current interaction data or learning history. These techniques enable the dynamic adjustment of the learning environment, facilitating personalized learning improvements through predictive modeling. *(2) Data mining techniques*: Data mining techniques: involve the processing, analysis, searching, and extraction of features using various analytical methods, such as clustering, multivariate analysis, and quantitative approaches, to derive insights from educational data. *(3) Recommendation techniques*: offer recommendations for relevant knowledge, course materials, problem-solving strategies, resources, and exercises tailored to the learner's progress and proficiency. *(4) Natural language processing (NLP) techniques*: are employed for syntactic and semantic analysis to comprehend, classify, and evaluate the feedback and needs of both learners and instructors. *(5) Expert knowledge*: as advisory tools, providing context-specific solutions and recommendations for both learners and instructors, based on individual circumstances and learning needs.

Recommender systems (RS) have emerged as a vital tool for processing end-user data. Particularly in addressing the challenges posed by vast volumes of information, diverse formats, and complex structures across various domains. Advances in RS techniques have revolutionized data utilization, enhancing efficiency in areas such as e-commerce, entertainment, and essential fields like education and training. One of the most popular approaches is collaborative filtering (CF), which generates recommendations based on

the behaviors and preferences of similar user groups from the past. This method assumes that users are likely to enjoy content favored by others with similar preferences. CF is typically categorized into two main types: *(1) User-based CF*: analyzes users' rating histories to identify "similar" users (measures such as Cosine, Euclidean, or Pearson). It then recommends items that these similar users have liked but the current user has not yet explored. *(2) Item-based CF*: evaluates the similarity between items based on how they have been rated or used by users. It then suggests items similar to those the user has previously liked or rated highly. Beyond CF, other techniques commonly employed in RS. Content-based filtering: recommendations are generated by analyzing the attributes of items. Knowledge-based filtering: this approach leverages domain-specific information and rules tailored to user needs or contexts. Hybrid methods: these combine multiple approaches to enhance accuracy and relevance. Ultimately, to optimize RS performance and enhance user experiences, the flexibility to integrate and adapt various methods remains a key factor across diverse applications.

In the context of ITS, session-based recommendation systems (SBRS) play a crucial role in providing accurate and timely learning suggestions. Instead of relying on past learning history, SBRS leverage students' behavior and interactions during each session. This enables personalized recommendations, thereby optimizing the learning experience. This approach is particularly important as students' learning objectives can change rapidly throughout the learning process. Numerous studies have focused on early prediction of learning outcomes based on data collected from ITSs. A study by Li [2] applied a 1D CNN (Convolutional Neural Networks) model to extract data features, combined with LTMS, to predict the course outcomes of students preparing for the start of a term. The experimental results were evaluated with an RMSE metric of 0.785. Further research by [3, 4] expanded this model by utilizing deep convolutional layers and bidirectional LSTM (Long Short-Term Memory networks) models to forecast complex phenomena such as gold price fluctuations and climate change over time. Additionally, Pan et al. [5] developed a diverse course RS that employs deep learning algorithms to extract features from educational data, thereby optimizing the curriculum for individual learners. A currently prominent research direction is the prediction of student performance particularly their ability to correctly respond on the first attempt to exercises posed by the ITS. This prediction not only helps assess the readiness of the learner but serves as the basis for optimizing teaching systems and assessments. The aim of designing curricula appropriately meets the specific needs of each student and enhance teaching effectiveness.

CFA is a key metric for assessing learners' knowledge readiness in ITS. This metric reflects the learner's ability to correctly answer a question or exercise on their first attempt. It also provides insight into their understanding and preparation for the learning content. CFA is widely used to predict learning outcomes in automated educational systems. It is integrated into ITS and plays a vital role in several areas. *(1) Assessment of learning ability*: Assessment of learning ability: CFA helps predict the likelihood of a learner answering a question correctly on their first attempt. It also forecasts overall learning performance throughout the course. Based on this information, the ITS can adjust question difficulty, ensuring an appropriate level of challenge that supports the learner's improvement. *(2) Personalization of learning progress and immediate feedback*: CFA allows the personalization of the learning path for each learner based on their CFA score. The system can provide advanced exercises for those with high CFA scores. For those with lower CFA scores, it can offer remedial lessons with instant feedback to help those with lower CFA scores improve. *(3) Adjustment of teaching strategies*: CFA enables the ITS to automatically recognize and adjust its teaching strategies. It does so by suggesting relevant lessons or materials that support the learner's progress, ensuring that the learning path is aligned with their abilities. *(4) Data analysis*: The effectiveness of lesson design, exercises, and questions is evaluated through CFA. A low CFA value may indicate that the content is too challenging, not aligned with the required knowledge, or lacks sufficient grounding. This can lead to gaps in the learner's problem-solving ability.

Building on our previous work presented at the FDSE 2023 conference [6], we introduce the ensemble CL-PSP model, which combines CNN and LSTM. This model is designed to predict learners' performance by analyzing their ability to correctly answer questions posed by ITS. To evaluate its effectiveness, we utilized two widely recognized educational datasets: KDDCup 2010 and Assistment 2017. The key contributions of this study include:

(1) Introduction of the CL-PSP model: We propose CL-PSP, a hybrid architecture that integrates static feature extraction within learning sessions using CNN and captures temporal dynamics through LSTM networks. This combination allows CL-PSP to accurately predict the CFA, an important indicator of a learner's ability to answer correctly on the first try. CFA has significant practical implications for educators and educational administrators, as it supports more effective planning, design, and adaptation of instructional activities.

(2) Demonstration of CL-PSP's superior performance: On the KDDCup 2010 dataset, the CL-PSP model achieved

a 2.60% reduction in RMSE compared to TAGNN_CFA. Although the results on Assistment 2017 were less pronounced, the RMSE of CL-PSP was approximately 5.47% higher than that of TAGNN_CFA, indicating potential for improvement. Additionally, our previously published LSTM-based model showed slightly inferior performance, with an RMSE increase of about 15.20% and 0.63% compared to CL-PSP, respectively. These results confirm the effectiveness and practical potential of CL-PSP in ITSs, particularly for tasks involving learning performance prediction through comprehensive comparative analysis.

II. RELATED RESEARCH AND RECOMMENDATION SYSTEMS

1. Related works in predicting academic performance

In the context of modern education, data mining has become an essential tool for enhancing learning efficiency and improving educational quality. Numerous studies have applied advanced approaches such as machine learning, neural networks, and hybrid models to predict learning performance, analyze behavior, and optimize learning processes. This study categorizes and highlights notable research according to these approaches, thereby emphasizing their practical applications in the educational domain.

(1) Machine learning approaches: Classical methods such as Decision Trees, KNN, SVM, XGBoost, LightGBM, Random Forest, and ensemble techniques have been widely used to predict CFA and learning performance from static interaction data in ITS and LMS platforms. BKT [7] and MF [8] proved effective in capturing knowledge accumulation. To improve classification, Tariq et al.[9] tested oversampling techniques, with KNN achieving the best result (83.7%) on a multi-class LMS dataset. While [10] combined SVM and ANN with an optimization method to predict pass/fail outcomes and final scores. These approaches highlight the role of preprocessing and feature selection in boosting performance on static data. The research [11] used Logistic Regression, Random Forest, and Naive Bayes to predict graduation GPA from semester-wise data of 357 students. Accuracy, precision, and F1-score improved in later semesters (up to 85%) as learning data accumulated, while early predictions were less reliable due to limited evidence. Though effective for cumulative prediction, the approach oversimplifies temporal variation and early learning progression. Hoq et al.[12] introduced an explainable ensemble model using KNN, SVM, and XGBoost, achieving RMSE = 0.151 on programming data. While SHAP enhanced interpretability, these models lack sequential modeling. Our CL-PSP combines sophisticated deep learning techniques to predict continuous learning behaviors and analyze time-series data, offering the ability to personalize predictions for individual learners.

(2) Neural network approaches: Recent deep learning approaches have explored various architectures to model student behavior and enhance predictive accuracy. STR-SA [13] applied an attention-based network to recommend discussion threads in MOOCs, capturing session-level interactions but lacking cumulative learning context. ClickTree [14] combined BiGRU with a tree-based decoder to model short-term clickstream behavior while producing interpretable predictions (AUC = 0.788), yet it focused only on within-session data. Wang et al. [15] extended the Deep Knowledge Tracing framework by incorporating fatigue-related features-such as session duration and activity intervals-into an RNN-based model to capture cognitive state transitions. While it improves intra-session prediction, the model relies heavily on accurate timestamp features and lacks multi-session generalization. DSMKT [16] introduced a dual-branch architecture combining masked self-attention and GRU, further enhanced with online knowledge distillation. It achieved strong results on multiple datasets (AUC = 0.8378, ACC = 0.7827 on ASSISTments09), though it still focuses primarily on next-step prediction without long-range learner modeling. GGCNN [17] employed Graph Convolutional Networks with Genetic Algorithms to model structured student-course relationships, achieving up to 98% accuracy and F1 = 0.84. Meanwhile, MixerKT [18] used a pure MLP-based architecture to model learning sequences efficiently, reaching AUC = 0.7575 and KC accuracy = 0.7171 on AL2005. However, both approaches lack sequential personalization over time. While each of these models contributes uniquely-whether through attention, RNNs, GCNs, or MLP-based designs-they often fail to capture long-term behavioral trends across sessions. Our proposed CL-PSP model addresses these limitations by integrating CNN and LSTM to jointly learn local patterns and long-range temporal dependencies, enabling personalized and time-aware CFA prediction.

(3) Hybrid models: Hybrid architectures combine complementary neural components to model complex educational and prediction tasks. *(i) Temporal–structural modeling*: Several studies incorporate graph-based representations into sequential learning frameworks. CLGT[19], a graph transformer model that leverages heterogeneous student interaction graphs and temporal position embeddings for predicting performance in collaborative learning. CLGT outperformed GCN, GAT, and MLP on two real-world datasets, but its effectiveness depends heavily on graph quality and scenario-specific generalization. GC-LSTM[20] predicted dynamic links by feeding graph snapshots into an LSTM, effectively capturing temporal dependencies

and structural transitions. [21] applied a similar model to traffic forecasting, reporting reduced MAE and RMSE. However, these models often face challenges when applied to sparse graphs or rapidly evolving network structures. *(ii) Temporal–spatial modeling*: CNN-LSTM hybrids are widely used in time-series tasks with both spatial and temporal features. CNN-LSTM model applied by [22] successfully predicted indoor temperature spikes, achieving a MAPE between 0.89% and 5.46%. Alhussein et al. [23] used the same architecture for household load forecasting with high short-term accuracy. [24] used the same architecture for household load forecasting with high short-term accuracy. In education, [24] leveraged CNN-LSTM to extract key performance indicators from LMS interaction logs, outperforming models. A recent study by Alnasyan et al. [25] benchmarked several deep learning architectures—including CNN, RNN, LSTM, Bi-LSTM, and GRU-on student performance prediction tasks using LMS activity logs. Among all models, BiLSTM achieved the highest performance with accuracy of 90%, outperforming CNN by 4.7%. The study highlight the importance of bidirectional temporal modeling in capturing behavioral patterns from student log data. Despite strong performance, these models often require high computational resources and struggle with large-scale or noisy data. Deeper architectures are also prone to overfitting when applied to imbalanced datasets. *(ii) Temporal–semantic modeling*: To capture semantic context in educational data, LBKT [26] integrated BERT embeddings with LSTM and Rasch-based encoding, improving AUC and memory efficiency on long sequences. Building on this direction, [27] proposed MCQStudentBERT, a transformer-based model that incorporates student history and question-answer semantics to predict answer choices. Their concatenation-based variant achieved (0.579) and F1 score (0.740) using Mistral 7B embeddings, outperforming summation-based approaches. [28] explored large language models for learner performance prediction by fine-tuning GPT-2 on sequences of question content, options, and correctness. On the ASSISTments 2015 dataset, GPT-2 (medium) achieved an AUC (0.83) and accuracy (81%), outperforming DeepIRT, a deep neural variant of item response theory, and DKVMN, a memory-augmented model for multi-skill tracing. TAGNN-Lazy[29], an extension of the original TAGNN model[30], improves performance in session-based product recommendation by using directed graphs and GNNs with attention to capture key user interactions. Evaluated on RSC15 and Diginetica, it outperforms baseline models in Precision@20 and MRR@20.

While effective in modeling long-range semantic and temporal patterns, these models incur high computational costs, limited generalizability, and reduced interpretability. Although each hybrid strategy targets specific aspects, challenges remain: structural models struggle with sparse and dynamic graphs; spatial models are resource-intensive and sensitive to noisy data; semantic models trade scalability and transparency for expressiveness. To address these limitations, the proposed CL-PSP integrates CNN for local feature extraction and LSTM for temporal dynamics, enhancing CFA prediction in ITSs.

2. Models in recommendation systems

LSTM is a type of RNN specifically designed to process and predict sequential or time-series data. The strength of LSTM lies in its ability to retain information over extended time periods. It addresses the issue of long-term dependencies and overcoming the vanishing gradient problem typically encountered in traditional RNNs [31]. Each LSTM layer is capable of preserving information from previous computations through its unique structure. An LSTM layer (Figure 1) comprises multiple cells, each containing three essential gates [32]. These gates are critical for controlling the flow of information through the network over time. They act as mechanisms to open or close pathways. This enables the network to decide whether to retain, add, or discard information in the memory state. This is achieved through the sigmoid activation function and matrix multiplication operations. LSTM networks include three primary gates. (1) Forget Gate (f_t): this gate determines which portions of information from the previous hidden state (h_{t-1}) should be discarded. Here, (x_t) represents the current input, and σ denotes the sigmoid activation function. (2) Input Gate (i_t): this gate decides which new information will be added to the memory state. It also updates the candidate memory state $\widehat{C_t}$. (3) Output Gate o_t: this gate determines which parts of the current hidden state h_t should be used to generate the output. It also updates the current memory state C_t.

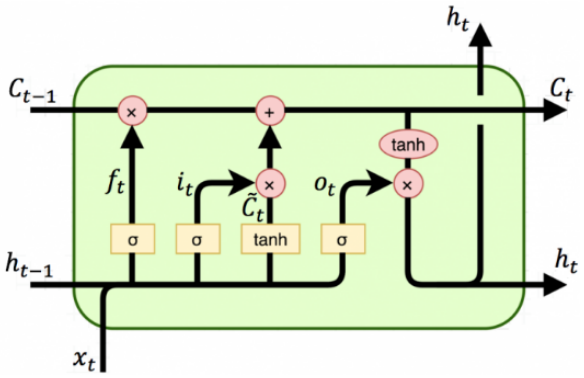

Figure 1. LSTM memory structure.

The proposed CL-PSP model is implemented with three LSTM layers, measuring the discrepancy between the actual labels and the model's predictions in binary classification tasks using the binary_crossentropy loss function (Eq.1.). This function optimizes the model by calculating the loss, the difference between the true labels (y) and the predicted labels ($\hat{y_i}$), to minimize the loss and improve accuracy. Here, (y) represents the ground truth label (a value of 0 or 1), and ($\hat{y_i}$) denotes predicted probability, typically obtained through the sigmoid activation function. The function [Eq.2.] transforms the output (s) into a probability value within the range [0,1].

$$Loss = -y_i log(\hat{y_i}) - (1 - y_i) log(1 - \hat{y_i}) \quad (1)$$

$$\hat{y_i} = \frac{1}{1 + e^{-s_i}} \quad (2)$$

Adam (Adaptive Moment Estimation) [33] is an optimization algorithm designed for rapid convergence by dynamically adjusting the learning rate for each parameter based on gradient information. It ensures a stable training process, particularly for sequential or complex time-series data. Adam helps fine-tune the weights during training to effectively minimize the loss function.

CF is a technique that can be used to predict learning outcomes, recommend appropriate learning methods, and analyze the progress of learners based on data from students with similar learning behaviors. According to Breese et al. [34], CF can be categorized into two main approaches: Model-based and Memory-based. *(1) Model-based method*: this approach employs machine learning algorithms to create prediction models using user and product data. For instance, techniques like MF or Singular Value Decomposition can be applied to uncover hidden relationships between factors that influence learning ability. These factors include test results, study habits, and classroom participation. This model can predict a student's scores on upcoming tests based on similarities with other students' learning characteristics. It effectively handles sparse data, scales well, and uncovers latent relationships within the data. *(2) Memory-based method*: this is the traditional approach in CF. Predictions about user preferences or behaviors are made by directly calculating similarities between users or products based on historical behavior data, without the need for machine learning models. Memory-based CF includes two main types: User-based CF and Item-based CF.

According to [35], CF considers the values users assign to products as linked features, forming the basis for filtering large datasets and supporting predictions and recommendations. CF is useful in three key tasks. (1) Predicting learning outcomes: Estimating students' scores or academic performance based on the learning behaviors of students with similar preferences or results. (2) Recommending learning methods: Suggesting study methods, learning materials, or lessons based on the study habits and performance of similar students. (3) Analyzing and predicting learning progress: Forecasting students' future progress or changes in performance, based on past data and the learning behaviors of other learners. Fundamental formulas in CF include: *(1) User Mean* (Eq.3.): the average value of the user rating scores assigned to the product is calculated. The average helps to balance the bias in user ratings, particularly when there are significant differences in rating levels. For $Rate_{u,i}$: the rating given by user u to item i; $Item_u$: the set of items rated by user u. $\overline{Mean_u}$: the average rating of u.

(2) Pearson Correlation [36]: it measures the similarity between two users based on their ratings of common items. It determines the degree of association between users through their ratings of similar products (Eq. 4). Here, $i \in Item_{u,v}$ refers to the set of items that both u and v have rated; $Sim_{(u,v)}$ represents the similarity between users u and v. *(3) Prediction of Ratings* (Eq.5): the prediction estimates the score a user will assign to an item based on the similarity between users or items. This estimate is computed by aggregating the ratings of similar users, adjusted to account for variations in users' average ratings. Let $Pred_{u,i}$: denote the predicted rating of u on i; N refers to the set of neighbors v of u for i. The value of the adjustment coefficient k can be identified through experimental trials and performance evaluation, where various values are assessed to optimize the model's accuracy and prediction quality.

$$\overline{Mean_u} = \frac{1}{|Item_u|} \sum_{i \in Item_u} Rate_{u,i} \quad (3)$$

$$Sim_{(u,v)} = \frac{\sum_{i \in I_{uv}} (Rate_u - \overline{Mean_u})(Rate_v - \overline{Mean_v})}{\sqrt{\sum_{i \in I_{uv}} (Rate_u - \overline{Mean_u})^2 \cdot \sum_{i \in I_{uv}} (Rate_v - \overline{Mean_v})^2}} \quad (4)$$

$$Pred_{u,i} = \overline{Mean_u} + k \sum_{v=1}^{N} Sim_{(u,v)} \left(Rate_{v,i} - \overline{Mean_v} \right) \quad (5)$$

In our study, the User-based CF technique was employed, utilizing user similarity computed through the K-Nearest Neighbors (KNN) algorithm. This method was applied to estimate correlations and predict students' ability to answer questions.

MF: when approached mathematically, involves decomposing a large data matrix R into the product of two or more smaller matrices, P and Q. Each of these matrices,

has reduced dimensions, achieved through a lower−rank representation that retains the essential information. This enables the identification of hidden patterns or features within the data. This method is widely used in various fields such as recommender systems, prediction, and deep learning. The result of this factorization process is the connection of sub−matrices. The goal is to reconstruct the original matrix R from P and Q while minimizing the error, ensuring that R is more accurate [37]. Key operations include the Loss Function, which measures measures the discrepancy between the actual value r_{uj} and the predicted value $\hat{r}_{ui}$ (Eq.6). It also includes the optimization function based on Frobenius norm (Eq.7). Under the assumption that $R[m,n]$ represents the input matrix and $P[m,k]$, $Q[k,n]$ are the optimal sub−matrices. MF in RS utilizes stochastic gradient descent (SGD) to predict missing values in the rating matrix, thereby optimizing the objective function.

Algorithm 1 KNN Algorithm for Performance Student Prediction

Require: X_{train}: User–item rating matrix $[u,i]$; X_{value}: Values of training data $[0,1]$; Y_{pre}: User–item prediction vector;

N: User-neighbors list on item i; K: Number of nearest neighbors.

Ensure: $\hat{y}_{\text{pre}}$: Predicted value of user Y_{pre}.

Step 1: Initialization

$distance \leftarrow \emptyset$ ▷ A distance list between u and its neighbors

Step 2: Main Process

for $t = 1$ to $|X_{\text{train}}|$ **do**

(1) Computing distance

$d[i] \leftarrow similarity(X_{\text{train}}, Y_{\text{pre}})[i]$

$distance[i] \leftarrow X_{\text{value}}[i]$

(2) Sorting

Sort $distance$ in ascending order

$N \leftarrow$ first K elements of $distance$

(3) Prediction

$rating \leftarrow$ average distance of $N[i]$

$\hat{y}_{\text{pre}} \leftarrow$ highest $rating$ in N

end for

return $\hat{y}_{\text{pre}}$

$$\hat{r}_{ui} = P_u.Q_i^T = \sum_{k=1}^{K} P_{uk}.Q_{ik} \tag{6}$$

$$L = \frac{1}{2} \sum_{(u,i)\in R} (r_{ui} - \hat{r}_{ui})^2 + \frac{\lambda}{2}(\|P\|_F^2 + \|Q\|_F^2) \tag{7}$$

SGD operates as follows: Random initialization of P, Q. (2) Updating each vector p_i, q_j (rows of P, Q respectively) based on the partial derivatives of the gradient to reduce the slope of the prediction error relative to the actual values. The updating process (Eq.8, 9) continues until the objective function L converges, reaching its minimum value. Let η (0¡ η ¡1) denotes the learning rate.

$$p_i \leftarrow p_i + \eta.[2.(r_{ui} - \hat{r}_{ui})^2.q_j - 2\lambda p_i] \tag{8}$$

$$q_j \leftarrow q_j + \eta.[2.(r_{ui} - \hat{r}_{ui})^2.p_i - 2\lambda q_j] \tag{9}$$

CNNs [38]: stand as a highly efficient deep learning architecture widely employed in diverse domains, including computer vision and speech recognition. Through the implementation of convolution operations at diverse levels of granularity, CNNs demonstrate exceptional proficiency in extracting essential non-linear features for learning tasks. This characteristic substantially diminishes the dependence on manual feature engineering. As a result, CNNs are particularly well-suited for deployment in predictive systems and SBRS. LeNet-5 is a prevalent CNN architecture, comprising layers: Convolutional, pooling, and fully-connected layers.

(1) Convolutional layers: consist of input data channels (I). The feature maps (O) contain output matrices of neurons. Each neuron is derived from the application of a filter to a region of the input data in the preceding convolutional layer. The filters K are the convolution kernels with x, y representing the indices of the row and column. The calculation of the value for a recently introduced neuron, positioned at coordinates (i, j) within the feature map associated with the k^{th} filter, is determined by (Eq.10). This value results from convolution with the c^{th} input channel. H, V represent the horizontal and vertical directions of the 2D convolution kernel, while x, y denote the indices of the corresponding row and column in this kernel for channel c. Let $K[p,q,c]$ signify the weight of the neuron at position p, q in the k^{th} filter. And $I(i+p, j+q, c)$ is the neuron value at the position $i+p, j+q$ in the preceding input layer (c^{th} channel). Here, b is the bias of the c^{th} input. Denote (f) as the $Sigmoid$, $Tanh$, or $ReLU$ activation function, depending on the CNN model architecture.

$$O_{[i,j,c]} = f(\sum_{p=0}^{K_{x-1}} \sum_{q=0}^{K_{y-1}} (I_{[i+p,j+q,c]}.K_{[p,q,c]}) + B_{[c]}) \tag{10}$$

(2) Pooling layers: also known as subsampling layers, are usually positioned immediately after Convolutional layers. They decrease the number of neurons, diminish the size of subsequent feature maps, and preserve vital information

from the features. Two widely used subsampling techniques are Max Pooling and Average Pooling. In essence, the neurons in each feature map of these Pooling layers are calculated from and connected to the preceding convolutional layers. And (3) Fully-connected layers or dense layers: Each neuron within this layer is an combination of all neurons from the previous layer. The aim is to produce a comprehensive semantic information layer, encompassing essential predictions or classifications.

III. DATA DESCRIPTION

1. Similarities between SBRS and learner assessment problem

KDDCup 2010 and Assistment 2017 were the two datasets utilized in our experiments, sourced from different ITS. Each dataset comprises a large collection of records in numerical or textual format. Each record contains multiple data fields representing corresponding feature values. Each record represents a question answered by a learner, including whether the answer was correct or incorrect. It also contains other critical information such as problem, learner, time, response duration, and knowledge component (KC) related to the question (Table I). During the question-solving process, learners could request assistance or knowledge hints provided by the ITS. Finally, the learner's response concludes the session for the current question.

Table I
OVERVIEW OF THE DATASET EXTRACTED FROM ITSS

Row	1	...	n
User	User 1	...	User n
Problem	Problem 1	...	Problem m
Step	Step 1	...	Step 2
CFA	0	...	0
KC	KC1	...	KCn
Start time	Value	...	Value
...	...	...	...

According to [39], a session is an independent data unit, clearly defined by a start and end timestamp, and may or may not follow a sequential order. The sequentiality in the two datasets is captured through the chronological order of learners' interactions with the system, during answering process. In these datasets, each session is represented in a data entry format, reflecting an independent, sequential transaction. Each session includes information such as the learner's details, the problem, relevant knowledge components, timestamps, question responses, and other related attributes. Due to the nature of the datasets, learners may appear intermittently in additional exercises or across different session periods. To improve the accuracy of predictions

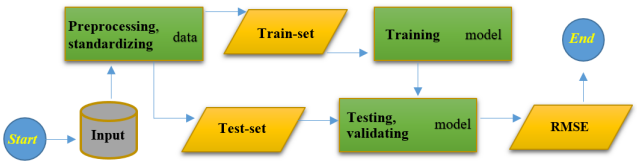

Figure 2. The approach to training and testing.

for the current step, we have organized the data sequentially based on the timing of each learner's interactions.

2. Data splitting method

In this educational setting, the main objective is to predict learning performance by forecasting the CFA value at the current step. This helps learners in mastering the necessary KCs. Our research experiments are focused on developing session-based data processing models to achieve this goal. The key features used in the training process include: (1) Student identifiers: learner identification. (2) Assignment identifiers: an exercise containing many questions. (3) Answer indicators: *CFA*, correctindicating whether the answer to the selected question is true or false. (4) Timestamps: start time, end time. Each row represents a single step, corresponding to a session, a response.

After preprocessing and normalization, the raw data undergoes feature extraction (Figure 2). To account for the temporal sequence, the datasets are organized by learners and the order of their question responses. This order serves as the grouping criterion. Each cohort of users is then split into two parts, with 70% assigned to the first segment and 30% to the last segment. These segments are aggregated to create the training set, comprising 70% of the source data. The remaining 30% is reserved for the testing set (Figure 3). The target value to be predicted is binary, where 1 indicates a correct response from the learner and 0 denotes an incorrect response to the respective question. The difference between the predicted and actual values is evaluated using the RMSE metric. The error evaluation process on the testing sets does not include the CFA field.

Each assignment will contain many independent and interrelated steps so that the final result can be obtained. Instructional items are related to questions and steps involved in problem-solving. An example of a problem: *"Calculate the radius of the circle with center O and radius R¨*. This is considered an exercise, where *"calculate the radius R"* becomes a question, a step. When the learner answers this question, the information is recorded as a session. Table II presents the general structure of the two training datasets and the features used to train our CL-PSP model. The KDDCup 2010 comprises 23 columns and 93,195 rows. Data from 70 learners who answered 6,666 questions

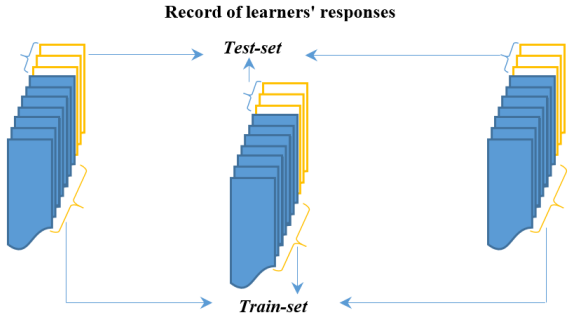

Figure 3. An overview of the train-set and the test-set [40].

were used across a total of 59,864 training sessions and 33,331 test sessions. In contrast, the Assistment 2017, obtained from 276 learners who solved 204 questions. It consists of a total of 231,403 rows and 76 columns. The corresponding training and testing sets contain 154,204 and 77,199 sessions, respectively.

Table II
DATASET ARCHITECTURE

Dataset		KDDCup 2010	Assistment 2017
Session count	Train-set	59,864	154,204
	Test-set	33,331	77,199
Features		23	76
Training features		5	5
Problems		6,666	204
Users		70	276
Average No. steps/learner	Train-set	855	559
	Test-set	497	424

Given the nature of the system, each learner engages differently when answering questions and exercises. Therefore, we estimate the average number of questions answered by each learner on the ITS. This helps support data observation and statistical analysis in the following section. This experimental version is based on the observations of characteristics recorded by ITS. Specifically, the number of interactions per user varies across the datasets. In KDDCup 2010, users are required to provide responses without the option to skip, whereas Assistment 2017 does not impose such a constraint. Consequently, we selected and processed data only for users with interaction counts ranging from 2 to 20 for the KDDCup 2010 dataset and from 2 to 50 for the Assistment 2017 dataset.

IV. EXPERIMENTAL DESIGN

To evaluate the effectiveness of the CFA prediction models, experiments were conducted on the two widely-used educational datasets. The models under comparison include User Average, CF, MF, LSTM, TAGNN_CFA and the proposed hybrid model CL-PSP. For fairness, each model was trained using carefully selected and optimized hyperparameters.

1. Data preparation

The student and problem identifiers were encoded using LabelEncoder, while one-hot encoding was applied to convert them into discrete integer indices or feature vectors, depending on the model used for training. The interaction sequences for each student were sorted chronologically. The dataset was split into 70% for training and 30% for testing (in Table II)

2. Baseline models and hyperparameters

The User Average model predicts the CFA of a student by calculating the average of their previous CFA values. This serves as a simple baseline for comparison with more complex models. CF, model adopts a user-based approach, where similarity between students is computed using the Pearson correlation coefficient. CFA is predicted based on a weighted average of the responses from the top $K = 20$ most similar students. The model input includes learner-problem matrix and uses historical CFA values to infer predictions. MF is implemented by learning two latent matrices representing users and items, respectively. The hyperparameters include as follow: latent dimension $K = 2$, regularization coefficient $\lambda = 0.1$, learning rate = 0.005, and and number of training epochs = 25. The model is optimized using gradient descent by minimizing the squared error between the predicted and actual values.

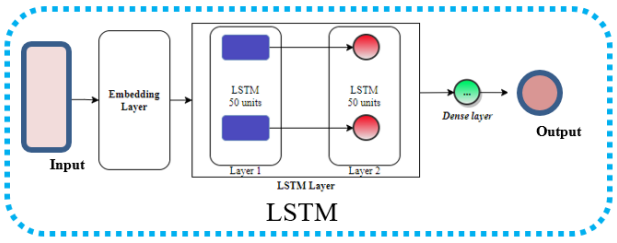

Figure 4. Prediction model from our previous work (FDSE 2023, [6]).

The LSTM model in Figure 4 is based on the architecture introduced in our previous study. It consists of two layers, each with 50 units, and a fully connected layer. This is implemented using the Keras library with the Sigmoid activation function. The model is trained using the Adam optimization algorithm, with a batch size of 1 and a learning rate of 0.0001. An early stopping mechanism is applied if the loss does not improve over 5 consecutive epochs. This

architecture has shown stable prediction results across both datasets, with low error and good generalization ability. TAGNN[30] is used as a baseline, adapted for CFA prediction by reformulating it as a binary classification model. TAGNN_CFA represents each session as a directed graph of learner interactions, with attention mechanisms highlighting the most relevant past actions. The model is implemented in PyTorch, trained with BCEWithLogitsLoss, and optimized using Adam. Hyperparameters include hidden dimensions (64-128), batch sizes (64–128), and learning rates (0.001-0.0005), with early stopping to prevent overfitting.

Building on this, our study proposes a hybrid CNN-LSTM model (CL-PSP) for predicting learner performance. CL-PSP combines the strengths of artificial neural networks to predict learners' academic performance. It does so across two distinct session-based datasets in the educational domain.

3. Model proposal

Figure 5 delineates the structure of CL-PSP, which integrates co-occurring single models in a time-sequential manner. The model targets the prediction of CFA values by leveraging temporal features, particularly the learner's question response start time and end time. Let t represent the current time step for prediction. While $(t-1)$, contextual features such as learner ID, exercise, number of steps, start time, and end time refer to the sequence of previous responses. These temporal signals are encoded and provided as inputs to both CNN and LSTM modules. The main objective is to use the time-based progression to determine whether the learner will answer the current question correctly. Training was conducted on the datasets with a maximum of 10 and 50 epochs, based on early stopping as no significant improvements were observed beyond these points. Hyperparameters were selected using RMSE as the primary evaluation metric. (1) The LSTM module is composed of three layers, each with 50 hidden units. The first two layers are configured with return_sequences=True to maintain temporal continuity, while the final layer outputs a tensor of size (None, 50). Dropout (0.5) is applied to reduce overfitting, and the model is trained using the Adam optimizer with a learning rate (0.001) and a batch size (32). LSTM layers {1,2,3,4} were applied in a grid search, with dropout rates ranging from 0.3 to 0.6, batch sizes {16, 32, 64}, and learning rates {0.01, 0.001, 0.0001}. This selected parameters consistently yielded the lowest RMSE values across experiments. (2) The CNN component uses 32 filters with a kernel size of 2 in its convolutional layer. These values were selected based on a grid search over candidate filter sizes {16, 32, 64} and kernel sizes {2, 3, 5}, aiming to optimize feature extraction and reduce

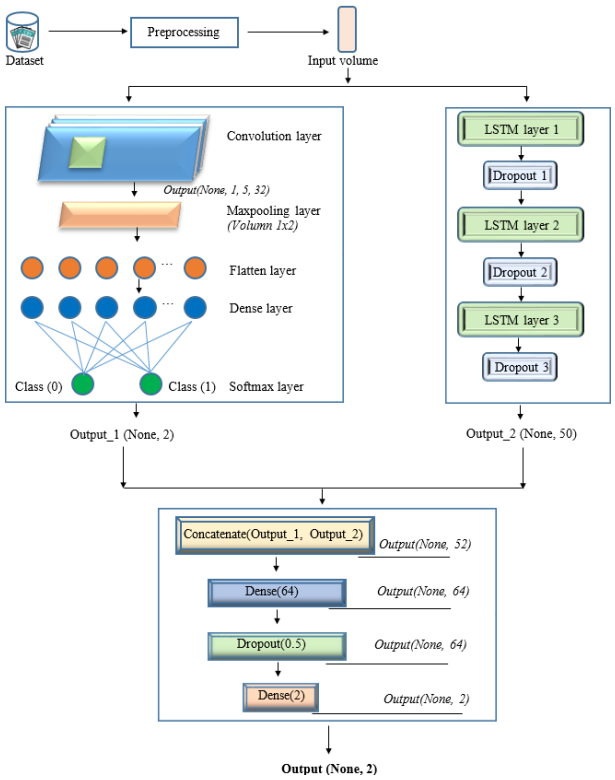

Figure 5. Proposed CL-PSP model.

RMSE. The convolutional output is then flattened into a one-dimensional vector and passed through dense layers for binary classification. (3) CNN-LSTM outputs are concatenated and processed through a shared dense layer with 64 neurons and a dropout rate of 0.5, which was found optimal for preventing overfitting without compromising generalization.

Our model proposed can be delineated into distinct learning and processing stages. Initially, the intricacies of the publishing model commence with the following stages. Data preprocessing involves eliminating outliers and selecting relevant features, resulting in the formation of input matrices. In the initial phase, the CNN module receives embedded input and performs spatial feature extraction through a convolutional layer. This process produces feature maps with reduced dimensions and ultimately generates a binary classification output {0,1}, representing the model's initial decision before fusion. LSTM receives input data and undergoes multiple layers to calculate temporal dependencies within the data. Layers retain crucial information from the past and maintain them over prolonged periods, facilitating the analysis and prediction of time series. Dropout layers are incorporated within the architecture to prevent overfitting and improve generalization. The final output of the LSTM module is a tensor of shape (None, 50),

representing the encoded temporal features. The final stage of training integrates the outputs from CNN and LSTM through a fusion layer, followed by normalization and classification steps. This results in a output of shape (None, 2), representing the predicted outcome. The proposed model offers enhanced prediction accuracy with reduced RMSE in comparison to state-of-the-art benchmark and other baselines. Notably, this hybrid model demonstrates a lower RMSE value, surpassing that of the model presented in our previous publication. The prediction algorithm implemented in our CL-PSP model is as follows:

Algorithm 2 Algorithm for Proposed Ensemble CL-PSP Model

Require: X: Input data; Y: True labels;
θ_{conv}: Parameters for convolutional layers; θ_{fc}: Parameters for fully connected layers;
θ_{lstm}: Parameters for LSTM layers; N: Number of training iterations.
Ensure: $\hat{Y}$: Predicted output.

Step 1: Initialization
Initialize parameters $\theta_{\text{conv}}, \theta_{\text{lstm}}, \theta_{\text{fc}}$

Step 2: Training Process
for $t = 1$ to N **do**
(1) Convolutional & LSTM Layers
$OutputL1 \leftarrow LSTM(X, \theta_{\text{lstm}})$
$OutputC1 \leftarrow Conv2D(X, \theta_{\text{conv}})$
$OutputL2 \leftarrow Dropout(OutputL1)$
$OutputC2 \leftarrow MaxPooling2D(OutputC1)$
$OutputL3 \leftarrow LSTM(OutputL2)$
$OutputC3 \leftarrow Flatten(OutputC2)$
$OutputL4 \leftarrow Dropout(OutputL3)$
$OutputC4 \leftarrow Dense(OutputC3, \theta_{\text{fc}})$
$OutputL5 \leftarrow LSTM(OutputL4)$
$OutputC5 \leftarrow Dense(OutputC4, \theta_{\text{fc}})$
$OutputL6 \leftarrow Dropout(OutputL5)$

(2) Concatenation
$OutputCL1 \leftarrow Concatenate(OutputC5, OutputL6)$
$OutputCL2 \leftarrow Dense(OutputCL1, \theta_{\text{fc}})$
$OutputCL3 \leftarrow Dropout(OutputCL2)$

(3) Predictions
$\hat{Y} \leftarrow Softmax(OutputCL3)$

(4) Compute Loss
$L \leftarrow Loss(\hat{Y}, Y)$
end for
return $\hat{Y}$

In essence, in the configuration for the proposed hybrid model, the initial LSTM architecture is a sophisticated arrangement featuring three LSTM layers, each housing 50 kernels. The configuration in the first two layers requires the use of $Return\_sequences = True$ to generate output sequences for each step in the input sequence, while the final LSTM layer only produces output at the last step. Meanwhile, the CNN incorporates 32 $cells$ for the first convolutional layer, employing overfitting techniques to control the number of trainable parameters. The data are flattened into a 1D vector before entering Dense layers, resulting in two output layers. The final stage in the above algorithm is the combination of the result data of the previous two models. This combined data continues to be learned in the Dense layer with $kernels$ (64), $dropout\ rate$ (0.5) to produce the final two classification results.

4. Assessment approach

RMSE [41] is a widely used metric for evaluating the accuracy of predictive models, particularly in the context of assessing learners' performance. In this scenario, the model predicts the probability of a learner answering a question correctly on their first attempt. RMSE (Eq. 11) measures the deviation between the model's predictions and actual outcomes. Its sensitivity to large errors makes it especially useful for identifying instances where the model's predictions significantly deviate from reality. Due to its mathematical properties, RMSE is commonly employed as a loss function in machine learning algorithms. Minimizing RMSE often corresponds to improving the model's predictive accuracy.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{11}$$

Let N represent the number of prediction samples, y_i the actual value, and $\hat{y}_i$ the predicted value. A smaller RMSE indicates that the model performs well, with predictions closely aligning with the actual values.

V. EXPERIMENTS

To assess the accuracy of the model and minimize the occurrence of overfitting, the Cross-validation method [42] is commonly used to evaluate the model's generalization ability, which refers to its capacity to perform well on unseen data. However, the characteristics of the datasets necessitate the use of the temporal sequence of learner responses. To address class imbalance and preserve the temporal structure, we adopted a user-based data partitioning

method that maintains the chronological order of learner interactions, rather than applying cross-validation

The data collection process from two different educational platforms has led to distinct characteristics that affect the model training and testing process. Data scale: KDDCup 2010 processes a significantly larger number of problems compared to Assistment 2017, while the number of questions is fewer. This difference can be explained by several interesting findings in each dataset, where the model also significantly influences the training process. (1) In KDDCup 2010, learners can skip questions without responding, whereas in Assistment 2017, every question must be answered before proceeding. (2) Consequently, the flexible navigation in KDDCup 2010 leads to longer sequences and more complex feature encoding, resulting in increased training time for CL-PSP.

In this experiment, CNN layers were integrated into LSTM layers with the aim of improving the prediction accuracy of learning ability. On the Assistment 2017 dataset, TAGNN_CFA achieved an RMSE (Figure 6) of 0.449, while our proposed CL-PSP model yielded an RMSE of 0.475. This indicates that TAGNN_CFA slightly outperformed CL-PSP, with an approximate 5.47% reduction in error. In contrast, on the KDDCup 2010 dataset, CL-PSP significantly outperformed TAGNN_CFA (Figure 7). The RMSE of TAGNN_CFA (0.385) was approximately 2.67% higher than that of the CL-PSP model (0.375) on the KDDCup 2010 dataset.

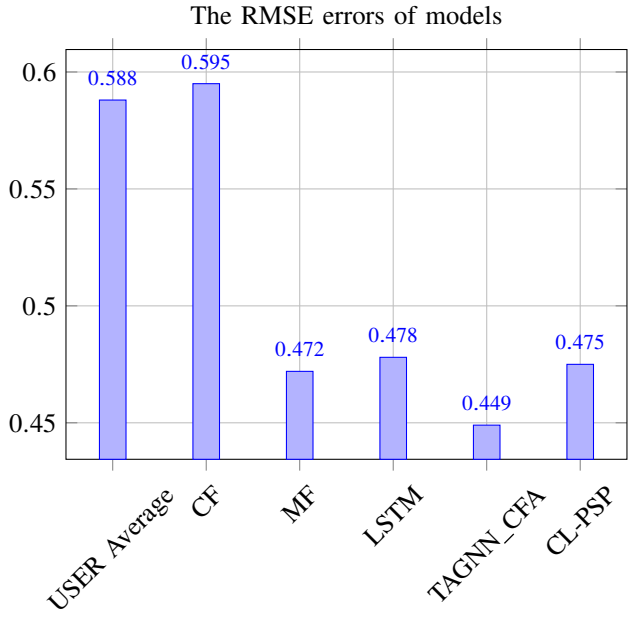

Figure 6. Assistment 2017 squared-error measurement.

This intriguing phenomenon can be explained by observing the structure of each dataset. In KDDCup 2010, 70

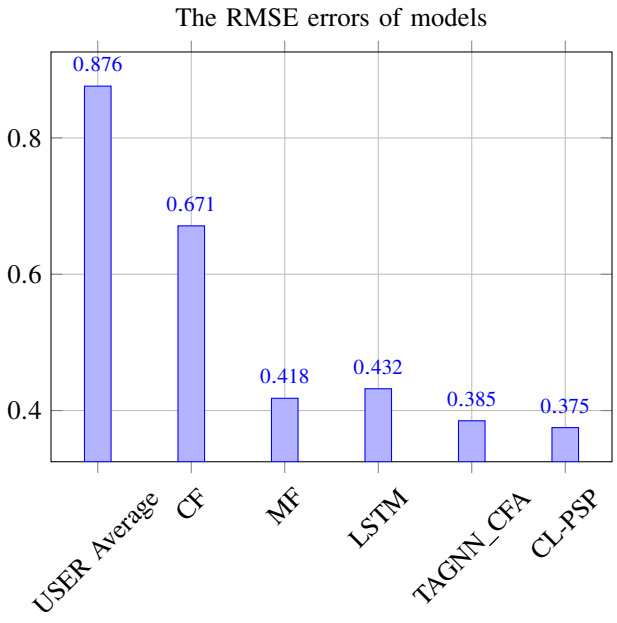

Figure 7. KDDCup 2010 squared-error measurement.

learners recorded an average of 855 sessions, suggesting a more focused and repeated engagement. In contrast, Assistment 2017 involved 276 learners with fewer interactions (559 sessions on average), leading to more variability and sparsity. These differences affect the learning patterns and subsequently influence model performance.

Our experiments aimed to reduce prediction error, with CL-PSP achieving notable RMSE improvement on the KDDCup 2010 dataset-highlighting the effectiveness of modeling sequential learning behavior. Although the improvement on Assistment 2017 was modest, the model remained stable and generalized well across diverse datasets.

Compared to the LSTM model [6], CL-PSP incorporates CNN to capture localized interaction patterns and LSTM to model temporal dynamics, enabling the model to detect both short-term fluctuations and long-term trends-thus improving CFA prediction accuracy.

In general, a consistent experimental setup across all baseline models confirms that the observed performance gains are attributed to the model architecture itself. CL-PSP further strengthens the comparative analysis by being benchmarked against an advanced attention-based model (TAGNN_CFA), as well as multiple traditional and modern baselines such as MF, LSTM, CF, and User Average. This approach (1) demonstrates CL-PSP's practical value in ITS by supporting personalized instruction through time-aware learner modeling, and (2) affirms the competitiveness and adaptability of CL-PSP across various data contexts.

VI. CLOSING REMARKS

The extension of our research focuses on integrating sequential time-series feature data and constructing a ensemble SBRS model with the aim of predicting learning performance. Experimental findings indicate that the proposed ensemble CL-PSP model exhibits enhanced performance in reducing RMSE errors, outperforming numerous baseline models. The analysis and experiments are conducted on two educational datasets. While the CL-PSP model achieves the best RMSE performance on the KDDCup 2010 dataset, the results on Assistment 2017 indicate that there is still room for improvement compared to MF.

Our research contributes to improving the performance of error prediction models applied in educational contexts. We further enhance the feasibility of implementing session-based data processing models, experience-based features and sequential characteristics into ITS. Moving forward, our investigation and exploration of educational datasets using advanced recommendation and prediction techniques are aimed to be continued. This endeavor seeks to bridge the gap between teaching and learning practices, thereby fostering advancements in the field.

Competing Interests: There are no competing Interests

Funding Information: There is no funding

Author contribution: The authors equally contribute to this work

Data Availability Statement: Not Applicable

Research Involving Human and /or Animals: Not Applicable

Informed Consent: Not Applicable

REFERENCES

[1] A. Yuce, A. M. Abubakar, and M. Ilkan, "Intelligent tutoring systems and learning performance: Applying task-technology fit and is success model," *Online Information Review*, vol. 43, no. 4, pp. 600–616, 2019.

[2] S. Li and T. Liu, "Performance prediction for higher education students using deep learning," *Complexity*, vol. 2021, no. 1, p. 9958203, 2021.

[3] X. Jin, X. Yu, X. Wang, Y. Bai, T. Su, and J. Kong, "Prediction for time series with cnn and lstm," in *Proceedings of the 11th international conference on modelling, identification and control (ICMIC2019)*. Springer, 2020, pp. 631–641.

[4] I. E. Livieris, E. Pintelas, and P. Pintelas, "A cnn–lstm model for gold price time-series forecasting," *Neural computing and applications*, vol. 32, pp. 17 351–17 360, 2020.

[5] X. Pan, X. Li, and M. Lu, "A multiview courses recommendation system based on deep learning," in *2020 International Conference on Big Data and Informatization Education (ICBDIE)*. IEEE, 2020, pp. 502–506.

[6] N. X. H. Giang, L. Thanh-Toan, and N. Thai-Nghe, "Session-based recommendation system approach for predicting learning performance," in *International Conference on Future Data and Security Engineering*. Springer, 2023, pp. 312–327.

[7] S. Pu, G. Converse, and Y. Huang, "Deep performance factors analysis for knowledge tracing," in *International Conference on Artificial Intelligence in Education*. Springer, 2021, pp. 331–341.

[8] T.-N. Huynh-Ly, H.-T. Le, and N. Thai-Nghe, "Deep biased matrix factorization for student performance prediction," *EAI Endorsed Transactions on Context-aware Systems and Applications*, vol. 9, 2023.

[9] M. A. Tariq, A. B. Sargano, M. A. Iftikhar, and Z. Habib, "Comparing different oversampling methods in predicting multi-class educational datasets using machine learning techniques," *Cybernetics and Information Technologies*, vol. 23, no. 4, pp. 199–212, 2023.

[10] M. Arashpour, E. M. Golafshani, R. Parthiban, J. Lamborn, A. Kashani, H. Li, and P. Farzanehfar, "Predicting individual learning performance using machine-learning hybridized with the teaching-learning-based optimization," *Computer Applications in Engineering Education*, vol. 31, no. 1, pp. 83–99, 2023.

[11] A. E. Tatar and D. Düştegör, "Prediction of academic performance at undergraduate graduation: Course grades or grade point average?" *Applied sciences*, vol. 10, no. 14, p. 4967, 2020.

[12] M. Hoq, P. Brusilovsky, and B. Akram, "Analysis of an explainable student performance prediction model in an introductory programming course," in *the 16th International Conference on Educational Data Mining (EDM 2023)*, 2023.

[13] M. Zhang, S. Liu, and Y. Wang, "Str-sa: Session-based thread recommendation for online course forum with self-attention," in *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2020, pp. 374–381.

[14] N. Rohani, B. Rohani, and A. Manataki, "Clicktree: A tree-based method for predicting math students' performance based on clickstream data," *arXiv preprint arXiv:2403.14664*, 2024.

[15] H. Wang, Q. Wu, C. Bao, W. Ji, and G. Zhou, "Research on knowledge tracing based on learner fatigue state," *Complex & Intelligent Systems*, vol. 11, no. 5, p. 226, 2025.

[16] Q. Ning, C. Guo, K. Liu, J. Tang, S. Zhang, W. Zeng, and X. Zhao, "Dual sequence modeling for knowledge tracing," *Data Science and Engineering*, pp. 1–12, 2025.

[17] T. Li, "Prediction and optimization of student grades based on genetic algorithm and graph convolutional neural networks," *International Journal of Computational Intelligence Systems*, vol. 18, no. 1, p. 59, 2025.

[18] J. Wang, M. Wang, Z. Li, K. Chen, J. Mei, and S. Zhang, "Mixerkt: A knowledge tracing model based on pure mlp architecture." *Computers, Materials & Continua*, vol. 82, no. 1, 2025.

[19] T. Peng, Y. Liang, W. Wu, J. Ren, Z. Pengrui, and Y. Pu, "Clgt: A graph transformer for student performance prediction in collaborative learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 13, 2023, pp. 15 947–15 954.

[20] J. Chen, X. Wang, and X. Xu, "Gc-lstm: Graph convolution embedded lstm for dynamic network link prediction," *Applied Intelligence*, pp. 1–16, 2022.

[21] Z. Wu, M. Huang, A. Zhao *et al.*, "Traffic prediction based on gcn-lstm model," in *Journal of Physics: Conference Series*, vol. 1972, no. 1. IOP Publishing, 2021, p. 012107.

[22] S. A. Yusuf, A. A. Alshdadi, M. O. Alassafi, R. AlGhamdi, and A. Samad, "Predicting catastrophic temperature changes based on past events via a cnn-lstm regression mechanism," *Neural Computing and Applications*, vol. 33, no. 15, pp. 9775–9790, 2021.

[23] M. Alhussein, K. Aurangzeb, and S. I. Haider, "Hybrid cnn-lstm model for short-term individual household load forecasting," *Ieee Access*, vol. 8, pp. 180 544–180 557, 2020.

[24] A. S. Aljaloud, D. M. Uliyan, A. Alkhalil, M. Abd Elrhman,

A. F. M. Alogali, Y. M. Altameemi, M. Altamimi, and P. Kwan, "A deep learning model to predict student learning outcomes in lms using cnn and lstm," *IEEE Access*, vol. 10, pp. 85 255–85 265, 2022.
[25] B. Alnasyan, M. Basheri, and M. Alassafi, "The power of deep learning techniques for predicting student performance in virtual learning environments: A systematic literature review," *Computers and Education: Artificial Intelligence*, p. 100231, 2024.
[26] Z. Li, J. Yang, J. Wang, L. Shi, and S. Stein, "Integrating lstm and bert for long-sequence data analysis in intelligent tutoring systems," *arXiv preprint arXiv:2405.05136*, 2024.
[27] E. G. Gado, T. Martorella, L. Zunino, P. Mejia-Domenzain, V. Swamy, J. Frej, and T. Käser, "Student answer forecasting: Transformer-driven answer choice prediction for language learning," *arXiv preprint arXiv:2405.20079*, 2024.
[28] S. P. Neshaei, R. L. Davis, A. Hazimeh, B. Lazarevski, P. Dillenbourg, and T. Käser, "Towards modeling learner performance with large language models," *arXiv preprint arXiv:2403.14661*, 2024.
[29] M. Thi Cam-Nhung, N. Thuy Anh, and N. Thai-Nghe, "Sequential recommendation using graph neuron networks," in *International Conference on Future Data and Security Engineering*. Springer, 2024, pp. 66–79.
[30] F. Yu, Y. Zhu, Q. Liu, S. Wu, L. Wang, and T. Tan, "Tagnn: Target attentive graph neural networks for session-based recommendation," in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 1921–1924.
[31] T. T. Dien, S. H. Luu, N. Thanh-Hai, and N. Thai-Nghe, "Deep learning with data transformation and factor analysis for student performance prediction," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, 2020.
[32] P. J. Gonçalves, B. Lourenço, S. Santos, R. Barlogis, and A. Misson, "Computer vision intelligent approaches to extract human pose and its activity from image sequences," *Electronics*, vol. 9, no. 1, p. 159, 2020.
[33] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[34] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *arXiv preprint arXiv:1301.7363*, 2013.
[35] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian informatics journal*, vol. 16, no. 3, pp. 261–273, 2015.
[36] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," *Noise reduction in speech processing*, pp. 1–4, 2009.
[37] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 1, pp. 1–24, 2010.
[38] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, 2021.
[39] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 130–137.
[40] J. Stamper and Z. A. Pardos, "The 2010 kdd cup competition dataset: Engaging the machine learning community in predictive learning analytics," *Journal of Learning Analytics*, vol. 3, no. 2, pp. 312–316, 2016.
[41] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.
[42] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *Morgan Kaufman Publishing*, 1995.

Nguyen Xuan Ha Giang is PhD student in Information Technology at Cantho University, Vietnam. Master degree in Information Technology in 2011. Lecturer in the Faculty of Information Technology at Cantho University of Technology, Vietnam. Research topics: Recommender Systems, Data Mining, Machine Learning, Student Modeling & Intelligent Tutoring Systems
Email: nxhgiang@ctuet.edu.vn

Lam Thanh Toan received Master degree in Computer Science in 2020. Lecturer in the Faculty of Information Technology at Cantho University of Technology, Vietnam. Research interests: Recommender Systems, Data Mining, Machine Learning, Student Modeling & Intelligent Tutoring Systems
Email: lttoan@ctuet.edu.vn

Nguyen Thai Nghe received the title of Associate Professor in October 2015. Dean of the Faculty of Information Systems, College of Information and Communication Technology, Can Tho University, Vietnam. Ph.D. in Computer Science (January 2009 – April 2012) at ISMLL, University of Hildesheim, Germany. Research topics: Recommender Systems, Data Mining, Machine Learning, Student Modeling & Intelligent Tutoring Systems, and Applications in Object Recognition.
Email: ntnghe@cit.ctu.edu.vn