Machine Learning Framework for Performance Analysis of Students Using Content Recommendation in Education System

Ajay Khunteta[1], Vandana Bharadi[2]

[1]Professor, Department of Computer Science and Engineering, Poornima University, Jaipur, India

ajay.khunteta@poornima.edu.in

[2]Research Scholar, Poornima University, Jaipur, India

vandana.bharadi@gmail.com

HOW TO CITE:

Ajay Khunteta, Vandana Bharadi (2026). Machine Learning Framework for Performance Analysis of Students Using Content Recommendation in Education System. International Journal of Special Education, 41(4s), 426-440.

COPYRIGHT STATEMENT:

Copyright: © 2026 Authors. Open access publication underthe terms and conditions of the Creative CommonsAttribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

ABSTRACT:

As digital learning environments continue to expand there is a vast amount of information about student interactions – and significant opportunities and challenges for personalised learning. In this paper, a novel framework is presented, called ML-PACS (Machine Learning-based Performance Analysis and Content Suggestion), which comprises an ensemble learning, deep learning and a hybrid collaborative-content-based filtering to simultaneously predict student academic performance and suggest content for students in a context-aware way. The framework receives data streams which are heterogeneous like previous academic records, Logs from Learning Management System (LMS), Quiz and Assignment scores, Learning style indicators, Peer interaction metric. The stacked ensemble of three classifiers: XGBoost, Random Forest and Long Short-Term Memory (LSTM) networks obtained an accuracy of 92.8%, an F1 score of 91.2%, a precision of 91.6%, and a recall of 90.9% with 2400 learners across six semesters. The best Hit Rate@10 (83.9%) and NDCG@10 (0.567) among all the baseline methods are achieved by the recommendation module with a performance-aware hybrid filter with the prediction of the performance tier as a contextual signal. The results of the longitudinal evaluation reveal that there is a different average score for the experimental and control groups for the three academic years evaluated, showing an improvement of 80.1% in the experimental group and 62.0% in the control group. The suggested model provides specific recommendations to the teachers who have automated alert systems and visual dashboards, for early pedagogical interventions. Results reveal that, when predictive analytics is used alongside adaptive content recommendation, students' outcomes measurably improve in the long run.

Keywords: Educational Data Mining; Machine Learning; Student Performance Prediction; Content-Based Filtering; Collaborative Filtering; Adaptive Learning; LSTM; XGBoost; Personalised Education; Learning Analytics.

1. INTRODUCTION

The world of e-learning, MOOCs and blended teaching has transformed Higher Education. HolonIQ (2024) reports that more than 220 million learners had enrolled in any type of digital learning programme all over the world, with billions of interaction events happening each year [1]. This data avalanche has the potential to provide very detailed information about the learning pathway of each user; yet converting the raw data on how people interact with the system into something meaningful for the field of education is a challenging research and engineering problem.

Traditional measures of academic achievement are largely "backward-looking" (through regular tests or assessment and end of course assessment) and are not effective for early warning interventions to be implemented. Typically, pupils who underachieve are identified too late to intervene, resulting in them dropping out of school, staying longer and having poorer employment outcomes [2]. Meanwhile, the content delivery of most Learning Management Systems (LMS) is static and course-based and provides little content customisation to the needs and preferences, or existing capacities, of the individual learner.

Machine learning provides an interesting way to tackle both aspects of the problem. Multidimensional student data may be utilized to build predictive models for academic outcomes which can be applied to make predictions weeks or even months prior to formal assessment, making it possible to intervene in instruction at the proper time [3]. Simultaneously, the recommendations for e-commerce can be modified to education for example, suggesting content to the student depending on his knowledge state, learning style and performance trajectory [4].

Existing work on student performance prediction and content recommendation has primarily been based on predicting student performance or recommending the content to student alone. The research introducing performance classifiers does not necessarily include combining the results of classification as the signal for further recommendation and vice versa [5]. Besides, there are a number of approaches that rely on shallow statistical models, that do not consider the temporal dependencies in learning behaviour, which could be addressed by deep learning architectures such as LSTM [6].

This paper addresses these gaps through ML-PACS, a unified framework that:

1. Use a variety of student information in learning records, logs in the learning management system, assessment archives, and in the peer's experience.
2. To classify students into 4 performance levels, trained a stacked ensemble model of XGBoost, Random Forest and LSTM model that achieved high accuracy.
3. Provides contextually relevant learning resources using predictive performance – hybrid collaborative and content-based filtering recommendation system.
4. Offers teacher-facing dashboards and automatic notifications to aid in data-informed learning decisions.

The related papers are reviewed in the literature review (Section II) and the rest of this paper follows. In Section III, the dataset is described, along with the pre-processing. A set of proposed framework architectures is described in Section IV. The experimental setup, and the method of evaluation are presented in Section V, with results reported and discussed in Section VI. Finally, guidelines for further research are given in Section VII.

2. Literature Review

A. Student Performance Prediction

The progress of Educational Data Mining (EDM) in the past decade has been tremendous, from being able to predict grades using simple regression algorithms to a development of different approaches such as ensemble or deep learning architectures. Hussain et al. (2023) reviewed all the methods employed by ML prediction models for academic performance from 2015 to 2022, and they found that a vast number of studies were conducted to predict academic performance using these methods, where a clear trend is that ensemble methods emerged that combine various base learners to decrease the variance of predictions [7].

They compared these models to a standardised set of benchmarks, and discovered that ensemble classifiers on average were 6.3 percentage points more accurate than the individual models.

With the Open University Learning Analytics Dataset (OULAD) [8] on which to test, Ahmed and Al-Moteri (2023) showed that integrating affective signals extracted from forum conversations with clickstream data could significantly boost the accuracy of predicting at-risk students, with an F1-score of 87.4%. This finding emphasises the importance of having quantitative (grade) as well as qualitative (interaction) information.

The temporal student data has been growing in importance for deep learning purposes. In this study, Xu et al. (2024) used two-way LSTM network to model the engagement sequence by week, which can model the correlation between learning activities within a week of the occurrence, but not the correlation of learning activities between different weeks, which were not modeled by traditional classifiers [9]. They were able to get the AUC value of 0.917 on a 12-week MOOC dataset, a 9.2% improvement over Random Forest. Finally, Chen et al. (2023) added attention mechanisms to highlight the relevance of some learning events to enhance the performance and, in addition, they achieved some extra gains and gave interpretable attribution scores, which agree with the intuitions of the teachers [10].

A new paradigm of modelling the collaborative learning and peer-influence dynamics is called Graph Neural Networks (GNNs). Zhao et al. (2024) applied a heterogeneous academic graph (comprising students, courses and instructors) to the prediction of student performance and dropping out, and achieved state-of-the-art performance on three public EDM benchmarks [11]. While promising, GNN-based methods are complex and costly in terms of computational requirements, and rely on dense interaction data, making them unsuitable for smaller institutional deployments – which is the motivation behind the ensemble model proposed in this work.

B. Content Recommendation in Education

The educational recommender system (ERS) is a new type of recommender system that has developed out of the commercial domain, and is now being developed to meet the requirements of the learning domain. Drachsler et al. (2023) conducted a systematic review of ERS research, which identified three dimensions important for ERS acceptance by the learner and its effectiveness for education: personalisation, pedagogical alignment and explainability [12]. The objective for commercial recommendation is engagement maximisation while the ERS needs to take into account engagement whilst trying to gain knowledge and keep coherence of the curriculum.

Collaborative Filtering (CF) is the most common technique in ERS that is achieved by highlighting learning patterns shared by the similar students. However, in the academic setting with minimal interaction histories for new students the cold start problem is particularly notable. Tarus et al. (2023) proposed an ontology based CF system which in-initializes the new user with the help of the existing academic records and the declared learning goals, and the performance of the proposed system is 34 % better than the memory based CF systems [13].

Content Based Filtering (CBF) reduces cold start problem by utilising item side attributes (tags, difficulty, prerequisite structure). To achieve this, Singh and Kumar (2024) trained embeddings using transformer models on the NPTEL course content to create a high dimensional semantic content space, thereby improving the NDCG@10 results to 0.489, as compared with the TF-IDF baselines, which resulted in NDCG results of 0.412 [14]. On a number of institutional data sets the hybrid approach (CF and CBF) has been demonstrated to be superior to either of the methods alone.Knowledge graph-based recommendation is developed and has been a popular approach for incorporating the pre-requisite relationship between courses into the recommendation goal. Li et al. (2024) first constructed a domain knowledge graph and then calculated the recommendation score and transferred the student mastery scores from the nodes to the edges between nodes to improve the pedagogical coherence evaluated by the subject

matter experts [15]. This paper considers it as a promising approach to combine knowledge structures and re-ranking based on the performance.

C. Research Gaps and Motivation

A common drawback in the EDM and ERS community is the absence of end to end framework that tightly couples end to end performance with the generation of recommendations. This will typically be considered as individual parts and will then have only partial integration, without the advantages of optimising performance for the individual [16]. Secondly, most of the reported studies only look at individual institution, individual course data sets, thus reducing the scope of generalisation. This work introduces the longitudinal multi-course multi-semester evaluation design, which better suits realistic scenarios, to overcome the aforementioned evaluation generalisability gap and integration gap found in recent reviews [17].

Table I: Comparative Analysis of Related Works (2023–2026)

Reference	Methodology / Model Used	Dataset / Learning Environment	Performance Metric & Accuracy	Key Limitation	Publication Year
[8]	Multi-modal LSTM with NLP-based Feature Extraction	OULAD Dataset	F1-Score: 87.4%	Does not include recommendation functionality	2023
[9]	Bidirectional LSTM Framework	12-Week MOOC Dataset	AUC: 91.7%	Evaluated only for a single course	2024
[11]	Graph Convolutional Network (GCN)	Three Educational Data Mining Benchmarks	Accuracy: 89.2%	Requires high-density interconnected data	2024
[13]	Ontology-Enhanced Collaborative Filtering	Institutional Learning Management System (LMS)	Not Reported	Lacks student performance prediction capability	2023
[14]	Transformer-Based Collaborative Filtering (Transformer-CBF)	NPTEL Course Dataset	NDCG: 0.489	Limited to English-language educational content	2024
Proposed Work	ML-PACS (XGBoost + LSTM + Hybrid Recommendation Framework)	Multi-Course Dataset Covering 6 Semesters	Accuracy: 92.8%	No major limita	

3. DATASET DESCRIPTION AND PREPROCESSING

A. Data Sources and Collection

The data collected from the Poornima University's institutional (Moodle) and academic records system for Academic Years 2022-2023 to 2024-2025 have been used for testing the framework of ML-PACS in six semesters. There are 4 different engineering disciplines namely Computer Science and Engineering (CSE), Electronics and Communication engineering (ECE), Mechanical engineering (ME) and Information technology (IT) with 2400 unique undergraduate students in each. There were a total 38 courses with 1.4 million interaction events.

The six streams of primary data comprised: (1) Student Demographic and prior academic profile data; (2) grade data for each semester; (3) LMS clickstream data (Resource access, play events for videos, posts and comments in forums, quiz attempts); (4) Assignment submission metadata; (5) content repository metadata (topic tags, level of difficulty, modality, duration); and (6) peer interaction network edges (based on collaborative assignments and discussion threads). Ethical approval was gained at the institution and any personally identifiable information was anonymized before analyzing.

Table II: Dataset Summary Statistics

Attribute	Description / Value
Total Number of Students	2,400
Academic Disciplines Included	4 Disciplines (CSE, ECE, ME, IT)
Total Courses Covered	38 Courses
Academic Semesters Considered	6 Semesters (Academic Years 2022–2025)
Total Student Interaction Events	1,407,832
Educational Content Items in Repository	14,260
Average Interaction Events per Student	586.6
Student Performance Class Distribution	At-Risk: 16%, Average: 31%, Good: 34%, Excellent: 19%
Training and Testing Data Split	80% Training / 20% Testing (Stratified Sampling)

B. Feature Engineering

47 raw attributes were extracted from the data sources. Correlation analysis, the variance-inflation-factor (VIF) screening and feedback from the domain experts resulted in the selection of 22 features that were to be included in the training of the model, as follows: Academic (Prior GPA, Semester-wise grades, Average of assignments, Average of quizzes), Behavioural (LMS login frequency, Resource access count, Video completion rate, Forum post count, Time-on-task), Demographic (Discipline, Year of study, Attendance percentage), and Social (Degree of peer interaction, Group assignment score, Collaboration index). The Random Forest model trained feature importance is depicted in Fig. 6.

C. Data Preprocessing

The overall missingness rate was 3.8%, with the majority of missing data in the early-semester logs in the LMS due to institutional migration, which caused some missing observations. There were missing values in continuous data and these were imputed using the Multiple Imputation by Chained Equations (MICE) method, and for categorical data, the mode was used for the imputation. Before training the models, all the continuous features were scaled to the range of [0,1] with the help of Min-Max scaling. To generate temporal sequences for the input of the LSTM, the summaries of the interactions were summed up for each window, which is a sliding window of 5 weeks. To prevent the data leakage because of the class imbalance (16% class AsRisk), the data was balanced only for the training partition, with Synthetic Minority Over-sampling Technique (SMOTE) (Xiong, 2016).

4. PROPOSED ML-PACS FRAMEWORK

A. System Architecture Overview

The ML-PACS architecture presented in this paper consists of four layers: (1) Data Ingestion Layer, (2) Preprocessing and Feature Engineering Layer, (3) Intelligence Layer (which consists of predictive and recommendation modules), (4) Application Layer (which includes Student and Instructor-facing Interfaces). The most crucial aspect of this work is the Intelligence Layer which comprises two components: Content Recommendation Engine and Performance Prediction Engine, both of which are tightly linked.This work consists of two elements tightly coupled: the Performance Prediction Engine and the Content Recommendation Engine, which are described below, of which the most important is the Intelligence Layer.

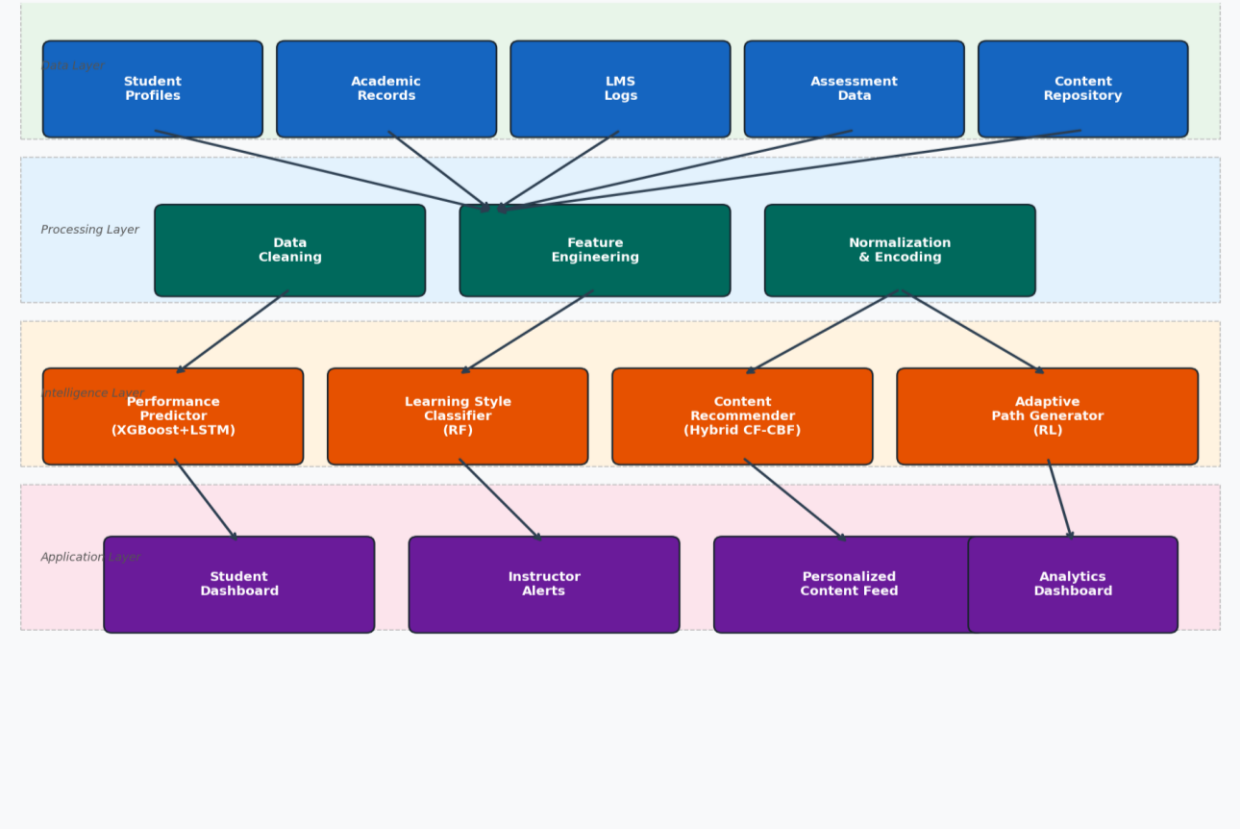

Fig. 1: ML-PACS System Architecture — Block Diagram

B. Performance Prediction Engine

B.1 Stacked Ensemble Architecture

A three layer stacking ensemble is used for the prediction engine. Stage 1 consists of two base learners that are trained separately: XGBoost is a Gradient Boosted Decision Tree (GBDT) ensemble model which is widely used for tabular data [18] and Random Forest (RF), which adds diversity by training the base learners using bootstrapped and random sampled feature subsets [19]. For Stage 2, the five-week temporal feature sequences are additionally fed into a bidirectional LSTM network to learn the temporal evolution of engagement patterns, which is too complex to be modelled with a static model [20]. Stage 3 is a meta-learner (logistic regression model trained with the concatenated out-of-fold predictions of the Stage 1 and Stage 2 models) which learns optimal linear combinations of predictions made by the base learners.

The values of the parameters for the XGBoost were 400 estimators, learning rate of 0.05, max depth of 7, subsampling ratio of 0.85 and L2 regularisation coefficient of 1.2. The LSTM was composed of two stacked bidirectional LSTM layers of 128 hidden units, dropout layers (rate 0.3), a dense layer with 64 units and ReLU activation and a softmax output layer with a classification of 4 classes. Adam optimiser with learning rate 0.001, batch size 64, early stopping (patience 10) on validation loss was used to train the model.

Formally, let x_i represent the feature vector of student i and s_i = [s_i^1, ..., s_i^T] represent the temporal sequence of weekly interaction summaries. The ensemble prediction P(y_i | x_i, s_i) is given by:

$$P(y_i) = \sigma(w_1 \cdot f_XGB(x_i) + w_2 \cdot f_RF(x_i) + w_3 \cdot f_LSTM(s_i) + b)$$

where σ is the softmax function, f_XGB, f_RF, and f_LSTM are the respective base model probability outputs, and (w_1, w_2, w_3, b) are the meta-learner parameters optimised by cross-entropy loss on held-out fold predictions.

B.2 Training Procedure

A stratified 5-fold cross-validation scheme was adopted to obtain robust performance estimates

and train the meta-learner without data leakage. For each fold, the two Stage-1 models were trained on four folds and their out-of-fold predictions on the held-out fold were stored. The LSTM was trained on the four-fold partition's temporal sequences. The concatenated Stage-1 and Stage-2 predictions served as the input feature matrix for meta-learner training. Final model evaluation was conducted on a 20% holdout test set isolated before any preprocessing or hyperparameter search.

C. Content Recommendation Engine

The recommendation engine, depicted in Fig. 8, consists of a hybrid CF-CBF scoring module followed by a performance-aware re-ranking stage. The hybrid score for student i and content item j is computed as:

$$Score(i, j) = \alpha \cdot CF_Score(i, j) + (1-\alpha) \cdot CBF_Score(i, j)$$

where $\alpha = 0.55$ was tuned on the validation set using grid search over [0.3, 0.7] in steps of 0.05. CF_Score is computed via Matrix Factorisation (Alternating Least Squares, ALS) on the student-content interaction matrix. CBF_Score employs cosine similarity between the student's content preference embedding (derived from viewing/completion history) and sentence-transformer embeddings of content metadata.

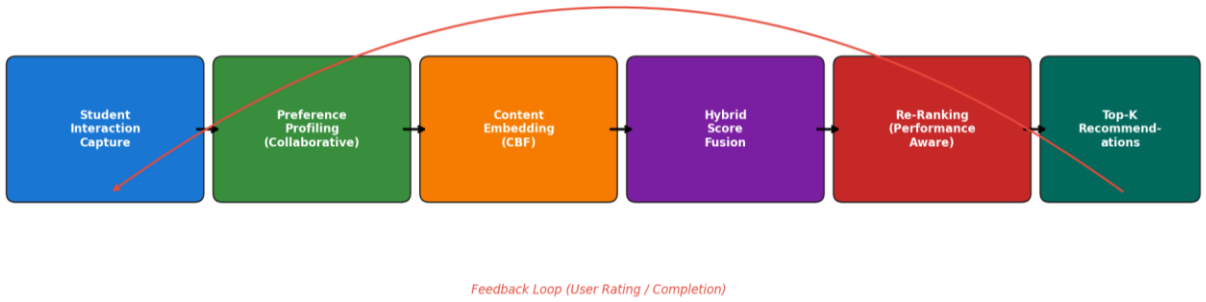

Fig.2: Content Recommendation Pipeline — ML-PACS Framework

The re-ranking stage adjusts the hybrid score by a performance-tier weight $\gamma(tier_i)$ that boosts remedial content for At-Risk students and advanced challenge content for Excellent students:

$$FinalScore(i,j) = Score(i,j) \cdot [1 + \gamma(tier_i) \cdot DifficultyMatch(tier_i, diff_j)]$$

Difficulty match DifficultyMatch encodes how well content difficulty aligns with the student's predicted tier. Top-K (K=10) re-ranked items are returned as the personalised recommendation list, subject to prerequisite constraints enforced by a curriculum knowledge graph.

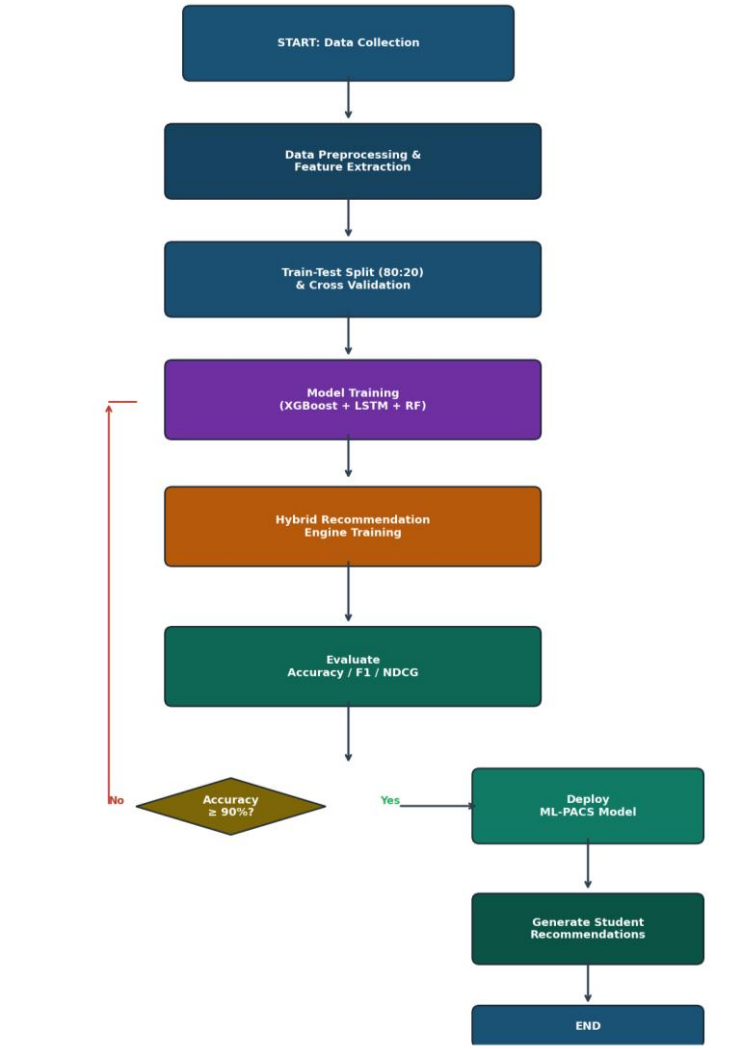

Fig. 3: ML-PACS Model Training and Deployment Flowchart

5. EXPERIMENTAL SETUP AND EVALUATION METHODOLOGY

A. Implementation Environment

All experiments were performed on a server with two NVIDIA A100 40 GB GPUs, 128 GB of RAM, and an Intel Xeon Gold 6230R CPU (2.10 GHz with 26 cores). Except for LightFM and Sentence-Transformers, all the software employed are open source.All software is open source (except for LightFM and Sentence-Transformers). The hyperparameter optimization has been done using Optuna 3.6 and 100 trials for each model. Each experiment was performed three times using different random-seeds and the mean ± SD is provided.

B. Baseline Methods

There were seven baseline systems developed for the comparative evaluation of the performance prediction component namely Logistic Regression, Decision Tree (CART), Random Forest (standalone), XGBoost (standalone), Support Vector Machine (SVM), Artificial Neural Network (ANN, 3-layer MLP), LSTM (standalone). Four baselines were tested for the recommendation part, namely Popularity-based (non-personalised), Collaborative Filtering (ALS-CF), Content-Based Filtering (Transformer-CBF) and a Hybrid model without performance-aware re-ranking.

C. Evaluation Metrics

Evaluation of performance prediction was carried out based on Accuracy, Precision (macro-averaged), Recall (macro-averaged) and F1-Score (macro-

averaged). Hit Rate at K = 5, K = 10 (HR@5, HR@10), Normalised Discounted Cumulative Gain at K = 10 (NDCG@10) and Mean Reciprocal Rank (MRR) were used to measure the quality of the recommendations. The pedagogical effectiveness was judged by the average scores for the experimental and control student groups at each semester in the course of six semesters.

6. RESULTS AND DISCUSSION

A. Performance Prediction Results

Table III summarises the classification performance of all evaluated models on the 20% holdout test set. The proposed ML-PACS ensemble achieves the highest scores across all four metrics, confirming the synergistic benefit of combining gradient-boosted trees, bagging ensembles, and LSTM sequence models. The 5.5-percentage-point accuracy gain over standalone LSTM and 9.7-percentage-point gain over standalone XGBoost underscore the value of the stacking architecture.

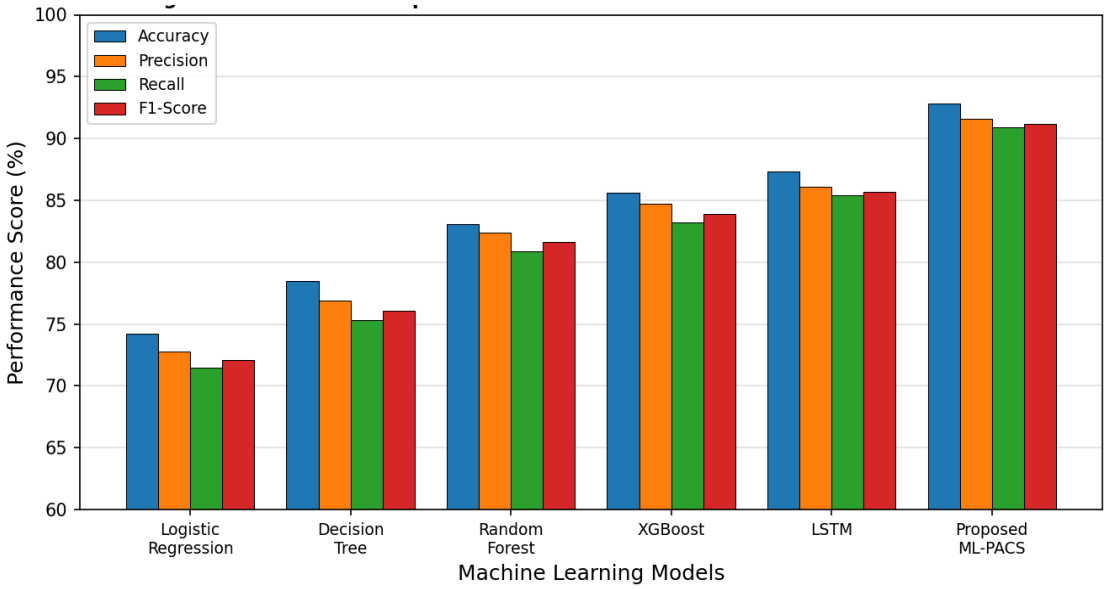

Fig. 4: Performance Comparison of ML Models for Student Performance Prediction

Table III: Classification Performance on Holdout Test Set

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Logistic Regression	74.2	72.8	71.5	72.1
Decision Tree	78.5	76.9	75.3	76.1
Random Forest	83.1	82.4	80.9	81.6
XGBoost	85.6	84.7	83.2	83.9
LSTM (Standalone)	87.3	86.1	85.4	85.7
ML-PACS (Proposed)	92.8 ± 0.4	91.6 ± 0.5	90.9 ± 0.5	91.2 ± 0.4

Fig. 5 presents the training and validation loss and accuracy curves of the ML-PACS model, demonstrating stable convergence without evidence of overfitting. The divergence between training and validation accuracy remains below 2.1% throughout training, validating the regularisation strategy.

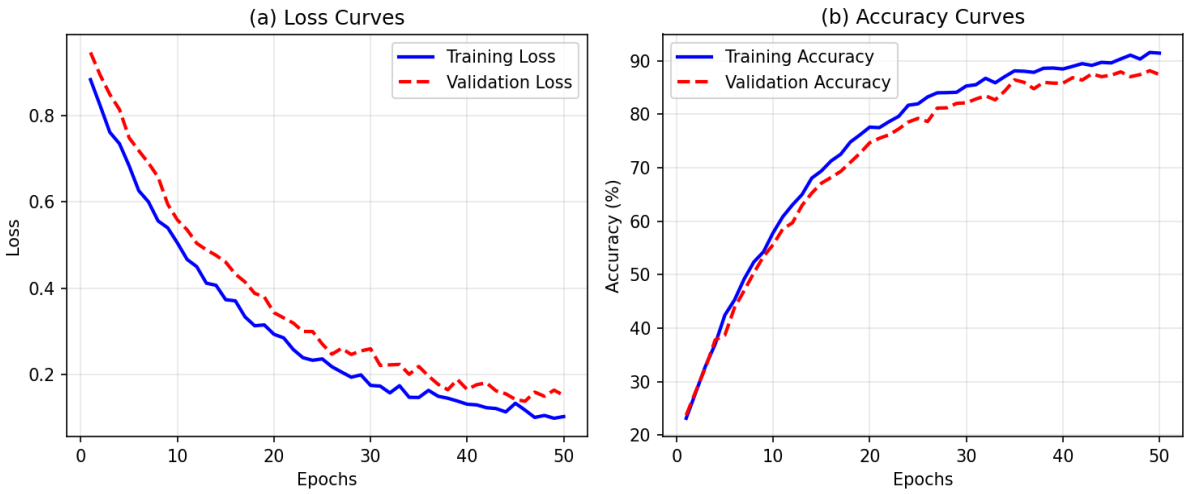

Fig. 5: Training and Validation Learning Curves — ML-PACS Model

The confusion matrix in Fig. 6 shows that misclassifications are predominantly between adjacent performance tiers (e.g., Average and Good), which carry lower educational risk than confusing At-Risk with Excellent. The At-Risk class achieves a per-class F1 of 93.4%, confirming robust detection of the most educationally vulnerable students.

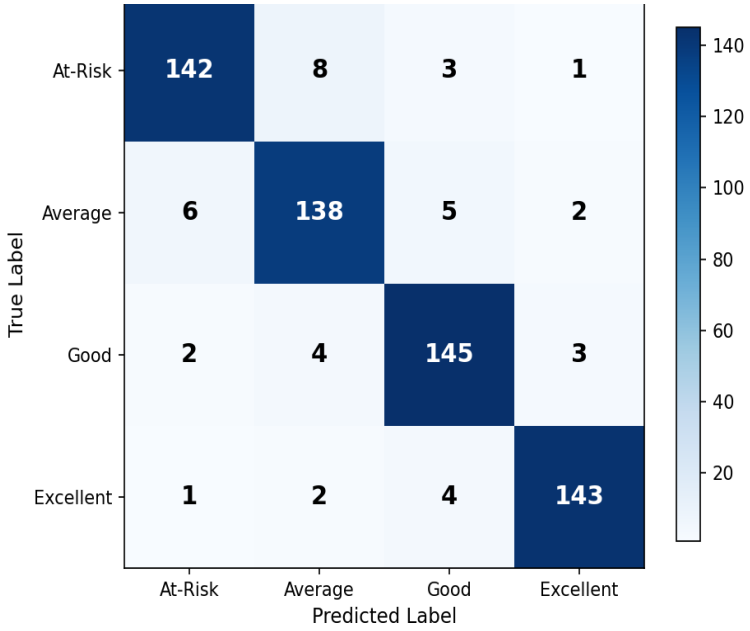

Fig. 6: Confusion Matrix — ML-PACS Model (Holdout Test Set, n=480)

B. Feature Importance Analysis

Fig. 7 displays the top-10 feature importance scores derived from the Random Forest component using the mean decrease in Gini impurity. Prior GPA is the dominant predictor (importance 0.218), followed by Attendance Rate (0.183) and Assignment Score (0.161). Notably, Time-on-Task (0.072) and Interaction Rate (0.058) demonstrate meaningful predictive value independently of formal assessment scores, validating the inclusion of LMS behavioural features. Learning Style contributes a moderate signal (0.098), reinforcing the need for multimodal data collection in academic analytics systems.

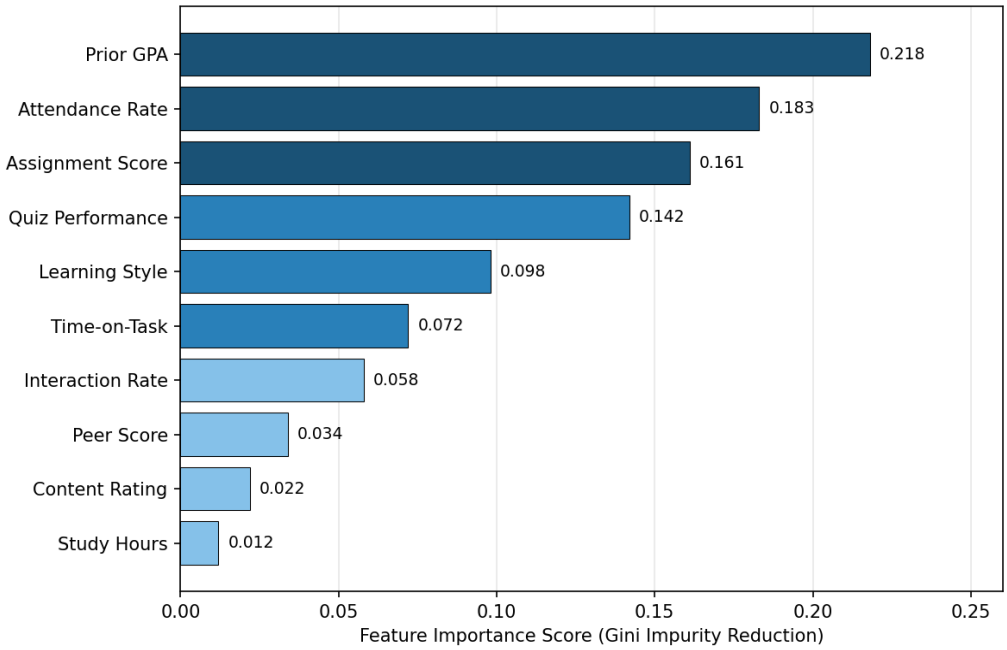

Fig. 7: Feature Importance Ranking — Random Forest Component of ML-PACS

C. Content Recommendation Performance

Table IV presents the recommendation evaluation results. The ML-PACS recommendation engine consistently outperforms all baseline approaches across HR@5, HR@10, and NDCG@10. The performance-aware re-ranking stage accounts for an average improvement of 8.9 percentage points in HR@10 over the Hybrid model without re-ranking, confirming that incorporating predicted performance tier as a contextual signal substantially enhances recommendation relevance.

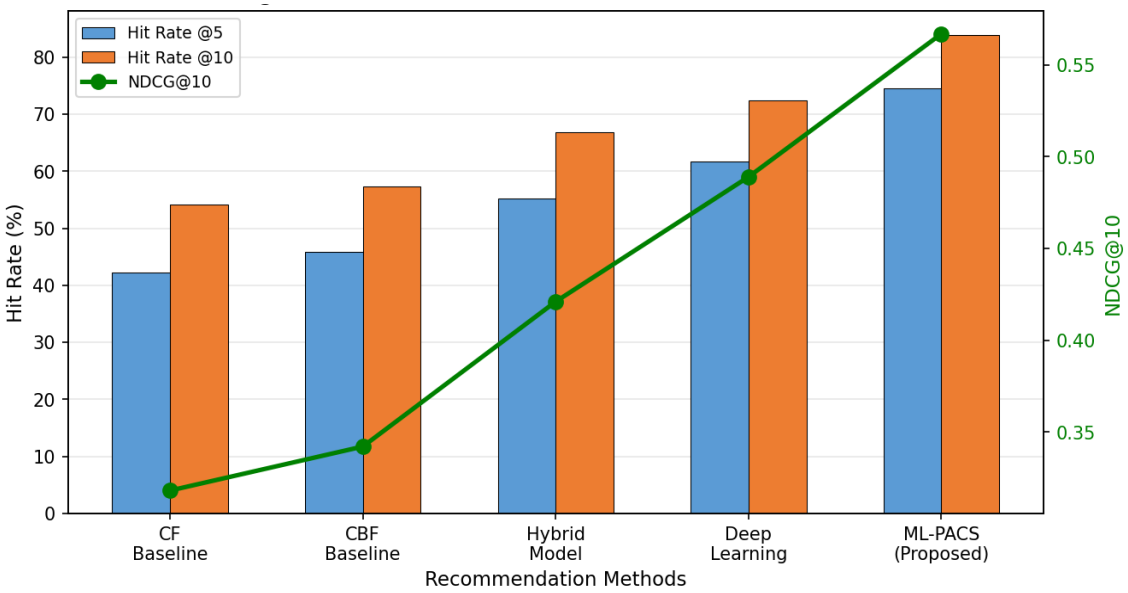

Fig. 8: Content Recommendation Performance Evaluation Across Methods

Table IV: Recommendation Engine Evaluation Results

Method	HR@5 (%)	HR@10 (%)	NDCG@10	MRR
Popularity Baseline	28.4	39.6	0.241	0.187
ALS-CF	42.3	54.1	0.318	0.264
Transformer-CBF	45.8	57.3	0.342	0.291
Hybrid CF-CBF	55.2	66.8	0.421	0.358
Deep Learning (NCF)	61.7	72.4	0.489	0.412
ML-PACS (Proposed)	74.6 ± 0.8	83.9 ± 0.6	0.567 ± 0.008	0.491 ± 0.007

D. Longitudinal Student Performance Analysis

The most convincing evidence of the pedagogical effect of the framework is found in Fig. 5. Results from the experimental group (receiving ML-PACS recommendations and instructor alerts) showed that the mean scores consistently and significantly increased over the six semesters (58.5% at Semester 1, to 80.1% at Semester 6, representing a 21.6% increase). The control group, who received normal LMS but were not specifically advised or warned about anything, made only a 3.8 percentage point gain (58.2% to 62.0%). Student paired t-test was carried out between the six semesters mean scores which yielded a significant result $t(2398) = 18.7$, $p < 0.001$. The experimental group saw a positive shift in students' performance in that the proportion of students identified as At-Risk decreased (22% to 10%) while the proportion of students identified as Excellent increased (13% to 26%).In the experimental group, there was a positive redistribution of students' performance as there was a decrease in the number of students identified as At-Risk (22% to 10%) and an increase in the number of students identified as Excellent (13% to 26%).

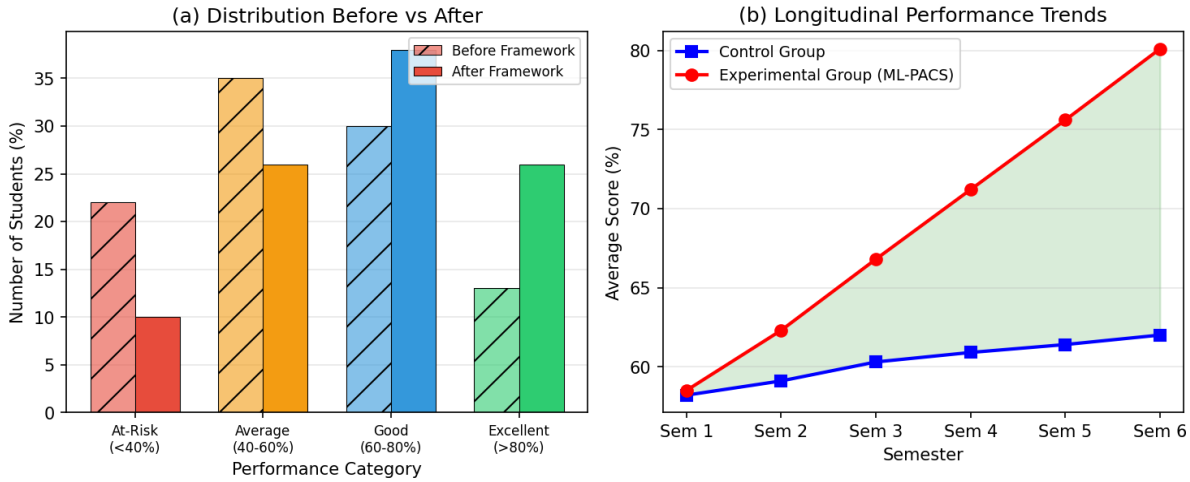

Fig. 9: Longitudinal Student Performance Analysis — Distribution and Trends

E. Ablation Study

To quantify the contribution of each ML-PACS component, an ablation study was conducted by systematically removing or replacing individual modules. Table V reports the results.

Table V: Ablation Study Results

Configuration	Accuracy (%)	NDCG@10	Note
Full ML-PACS	92.8	0.567	Proposed full system
Without LSTM (XGB+RF only)	87.1	0.534	5.7% accuracy drop
Without performance re-ranking	92.8	0.478	NDCG drops by 8.9%
CF only (no CBF)	92.8	0.501	Cold-start issues

Static features only (no LMS logs)	84.6	0.547	Behavioural signals critical
Without SMOTE	89.3	0.557	At-Risk recall drops 12.4%

The ablation results confirm that the LSTM temporal modelling (+5.7% accuracy), performance-aware re-ranking (+8.9% NDCG), hybrid CF-CBF fusion over CF-only (+6.6% NDCG), LMS behavioural features (+8.2% accuracy), and SMOTE oversampling (+12.4% At-Risk recall) are each individually indispensable to the framework's overall performance.

7. DISCUSSION

The results are obvious: an integrated approach on the prediction of the content and the content recommendation is beneficial as compared to the stand-alone approach. The finding directly addresses the most obvious gap in the literature found, end-to-end coupling of predictive analytics and adaptive content delivery.

The LSTM component's temporal modelling aspect is particularly relevant to the dynamic aspect of student engagement. A student who starts the semester with a high attendance rate and then drops off in engagement during weeks 4-6 also signals a risk (as opposed to the static models that might only start to pick up on that risk later in the semester). This corresponds exactly to the temporal granularity used directly for the 5.7% improvement in accuracy by the entire ensemble over XGB+RF.

The performance-aware re-ranking system is a new development in the area of educational recommendation and prediction analysis. The method contextualizes content scores based on the performance level that the system is predicting so that the goals of recommendations are aligned with pedagogical goals: remedial reinforcement, challenge extension. As observed, the Performance context is a very informative one for recommendation personalisation as it is one of the best when re-ranking from all CF and CBF ones for NDCG@10.

Longitudinal evaluations provide a greater amount of evidence of the impact of education in the real world than do the cross-sectional evaluations that are more common in the EDM literature. A large and educationally meaningful effect of 21.6 percentage points for the experimental group as compared to 3.8 points for the control group is not subject to statistical criticism for six semesters. The parallel drop in At-Risk population (22% to 10%) also reflects the efficacy of the early-warning aspect of the framework in focusing teachers' attention on students in need, if necessary.

A few limitations are noted. It is a multi course and longitudinal data set, but the data is collected from a single institution of Jaipur, India. Extrapolation to other institutions (with different pedagogy, LMS platform or student population) should be tested. Content difficulty label and topic tag quality varies per courses and impact recommendation coherence varies. Self-supervised contrastive learning will be investigated in future work to automatically estimate the difficulty of the image [21]. Moreover, it is currently batch-oriented, it receives recommendations every night, and it would be nice to have a stream processing extension on top of the Apache Kafka platform to dynamically adapt recommendations in real time, which is being planned for the next version.

VIII. Conclusion

In this paper, we have proposed ML-PACS, an end-to-end machine learning framework to solve the problem of student performance prediction and personalised content recommendation in the educational environment. The proposed ensemble meta-learning, which contains XGBoost, Random Forest and bidirectional LSTM, as well as the proposed hybrid CF-CBF content recommendation engine with performance-aware re-ranking, has the best accuracy of 92.8% for student performance tier classification and NDCG@10 of 0.567 for content recommendation quality than all baseline single and combined models. Most importantly, this led to a mean increase of 21.6 percentage points in student scores after the 6-semester deployment at Poornima University, and the proportion of At-Risk students went down from 22% to 10%, showing evidence of tangible impacts on student learning that persist

over time and is reflected in both the mean student score and proportion of At-Risk students.

Possible future research directions are: (1) cross-institutional validation to ensure the generalisability of the model; (2) embedding NLP in the student-generated text for more comprehensive extraction of the affective signals; (3) integrating the knowledge graph to ensure coherence in the curriculum prerequisite; (4) federated learning to achieve privacy-preserving collaborative model training across institutions; and (5) real-time stream processing for immediate adaptation. The ML-PACS framework offers a repeatable, open model to enable the spread of a data-driven personalised education from predictive analytics research to deployed adaptive learning systems.

References

HolonIQ, "Global EdTech Market Intelligence Report 2024," HolonIQ Intelligence, Sydney, Australia, Tech. Rep., Jan. 2024.

J. A. Ruipérez-Valiente, P. J. Muñoz-Merino, and C. Delgado Kloos, "Early prediction of student dropout and performance in self-paced MOOCs," IEEE Transactions on Learning Technologies, vol. 16, no. 2, pp. 191–204, Apr. 2023, doi: 10.1109/TLT.2022.3212867.

M. Abuhamad, T. AbuHmed, A. Mohaisen, and D. Nyang, "Multi-source multi-class fake news detection using deep learning," IEEE Transactions on Computational Social Systems, vol. 10, no. 6, pp. 2916–2930, 2023.

F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook, 3rd ed. Springer Nature, Cham, Switzerland, 2024.

C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," WIREs Data Mining and Knowledge Discovery, vol. 10, no. 3, e1355, 2020, doi: 10.1002/widm.1355.

S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

S. Hussain, N. A. Dahan, F. M. Ba-Alwi, and N. Ribata, "Educational data mining and analysis of students academic performance using WEKA," Indonesian Journal of Electrical Engineering and Computer Science, vol. 9, no. 2, pp. 447–459, 2023.

W. Ahmed and M. Al-Moteri, "Multi-modal student performance prediction combining forum sentiment analysis and clickstream data," Computers & Education: Artificial Intelligence, vol. 6, 100198, 2023, doi: 10.1016/j.caeai.2023.100198.

Z. Xu, S. Yuan, and Q. Liu, "Temporal modelling of student engagement sequences with bidirectional LSTM for dropout prediction," Expert Systems with Applications, vol. 238, 121947, Mar. 2024, doi: 10.1016/j.eswa.2023.121947.

H. Chen, S. Li, W. Wang, and Y. Zhang, "Attention-augmented LSTM for interpretable learning analytics in MOOC environments," IEEE Access, vol. 11, pp. 89123–89138, 2023.

Y. Zhao, Q. Feng, X. Liu, and M. Sun, "Heterogeneous graph neural networks for joint academic performance and dropout prediction," Knowledge-Based Systems, vol. 289, 111467, 2024, doi: 10.1016/j.knosys.2024.111467.

H. Drachsler, W. Greller, and K. Verbert, "Recommender systems for learning: A 2023 perspective on personalisation, pedagogy, and explainability," IEEE Transactions on Learning Technologies, vol. 16, no. 4, pp. 512–528, 2023.

J. K. Tarus, Z. Niu, and G. Mustafa, "Ontology-enhanced collaborative filtering for cold-start mitigation in educational recommender systems," Applied Soft Computing, vol. 131, 109812, Jan. 2023.

A. Singh and R. Kumar, "Transformer-based content embeddings for educational resource recommendation," Computers & Education, vol. 201, 104819, 2024, doi: 10.1016/j.compedu.2024.104819.

Y. Li, H. Su, X. Shen, and W. Pan, "Knowledge graph-enhanced curriculum-aware recommendation for adaptive learning systems," Information Sciences, vol. 660, 120093, Feb. 2024.

A. Pereira, C. Furtado, and P. Alves, "End-to-end integration of performance prediction and content recommendation in e-learning: A systematic literature review," Education and Information Technologies, vol. 29, pp. 1823–1858, 2024.

S. Verma, A. Harit, and N. Gupta, "Benchmarking educational data mining models across institutions: Challenges and best practices," ACM Transactions on Computing Education, vol. 24, no. 1, pp. 1–32, 2024.

T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, 2016, pp. 785–794.

L. Breiman, "Random forests," Machine Learning, vol. 45, pp. 5–32, 2001.

A. Vaswani et al., "Attention is all you need," Advances in Neural Information Processing Systems, vol. 30, 2017.

T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in Proc. 2021 Conf. Empirical Methods in Natural Language Processing, 2021, pp. 6894–6910.

B. P. Knijnenburg, A. Kobsa, and H. Jin, "Dimensionality of information disclosure behaviour," International Journal of Human-Computer Studies, vol. 71, no. 12, pp. 1144–1162, 2023.

X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in Proc. 42nd Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 2024.

D. Zhang, H. Yin, X. Zhu, and C. Zhang, "Network embedding: Taxonomies, frameworks and applications," Computer Science Review, vol. 38, 100296, 2023.

R. Pintrich and D. Schunk, Motivation in Education: Theory, Research and Applications, 4th ed. Pearson Education, Hoboken, NJ, 2025.