

Design and Implementation of an AI-Based Adaptive Learning System Using Bayesian Knowledge Tracing and Random Forest

Aparna Sikarwar

Department of Computer Science and Engineering
Raja Balwant Singh Engineering Technical
Campus, Bichpuri, Agra, Uttar Pradesh, India

Anshul Kumar Singh

Department of Computer Science and Engineering
Raja Balwant Singh Engineering Technical
Campus, Bichpuri, Agra, Uttar Pradesh, India

Alok Singh Jadaun

Department of Computer Science and Engineering
Raja Balwant Singh Engineering Technical
Campus, Bichpuri, Agra, Uttar Pradesh, India

Brajesh Kumar Singh

Department of Computer Science and Engineering
Raja Balwant Singh Engineering Technical
Campus, Bichpuri, Agra, Uttar Pradesh, India

Abstract

Traditional education systems teach all students the same content without considering individual knowledge levels or learning pace — a significant challenge in engineering education and self-paced online learning. We propose an AI-based adaptive learning system integrating Bayesian Knowledge Tracing (BKT) for real-time student modeling and a Random Forest classifier trained on the Open University Learning Analytics Dataset (OULAD) for performance prediction.

The system delivers personalized quizzes across four core engineering topics: Machine Learning, Data Structures, Databases, and Web Development. The Random Forest model achieved 90.9% classification accuracy, outperforming logistic regression by 14.7%. Behavioral engagement features accounted for 91.3% of predictive importance. A pilot study demonstrated 14.3% improvement in quiz accuracy and 16.1% increase in completion rates over a static quiz baseline. The system is lightweight, open-source, and deployable on a standard laptop without cloud infrastructure.

Keywords: *Adaptive Learning, Bayesian Knowledge Tracing, Random Forest, OULAD, Personalized Education, Student Modeling, Streamlit, Engineering Education*

1. Introduction

The dominant paradigm in formal education remains largely uniform — instructors deliver identical content to all students irrespective of prior knowledge or individual learning velocity. This is particularly ill-suited for engineering disciplines where prerequisite knowledge gaps compound rapidly. Adaptive learning systems offer a data-driven remedy by dynamically adjusting instructional difficulty and sequencing in response to observed learner behavior.

Bayesian Knowledge Tracing (BKT), introduced by Corbett and Anderson [8], provides a principled probabilistic framework for modeling the evolution of student knowledge over time. Ensemble classifiers such as Random Forest have demonstrated strong predictive performance on large-scale educational datasets, outperforming traditional methods on behavioral features [11][12].

Engineering education presents unique challenges: concepts are strictly hierarchical (e.g., Calculus → Differential Equations → Signals & Systems), and failure to master prerequisites leads to cascading difficulties. Traditional one-size-fits-all online courses cannot adapt to individual mastery gaps, causing high dropout rates in self-paced engineering programs. This paper directly addresses this problem by embedding a real-time mastery tracker (BKT) into a deployable quiz system that adjusts difficulty dynamically, ensuring students always practice at their zone of proximal development. This paper presents the design and

implementation of an adaptive learning system unifying BKT-based student modeling with Random Forest performance prediction. Key contributions: (1) a two-level

student intelligence model combining BKT mastery estimates with behavioral prediction; (2) a three-tier fallback question selection strategy; (3) multilingual feedback via Hinglish explanations; and (4) empirical validation through OULAD benchmarking and a controlled pilot study with M engineering students.

2. Literature Review

Martin et al. [1] reviewed 348 studies on technology-enhanced learning and found growing AI adoption while identifying a persistent gap in educator-focused tools. Zawacki-Richter et al. [2] observed that faculty are rarely involved in AI tool design for higher education. Continuous formative assessment has been advocated over summative evaluation as a more effective mechanism for tracking progress [3].

Pelaneck [4] identified the principal paradigms for student knowledge modeling — BKT, Reinforcement Learning, and Deep Neural Networks — noting trade-offs between interpretability and predictive power. Murtaza et al. [5] catalogued recurring challenges including cold-start problems, data privacy, and scalability limitations.

For predictive modeling, Kuzilek et al. [10] introduced the OULAD dataset comprising 32,593 student records. Jawad et al. [11] and Balabied et al. [12] confirmed that Random Forest outperforms SVM and Logistic Regression on OULAD, with behavioral features proving more informative than raw scores. Sari et al. [14] reported adaptive difficulty improves learning outcomes by 23% on average. Three unaddressed gaps exist: insufficient lightweight deployable systems, limited BKT-ML integration, and inadequate multilingual feedback support — all addressed by this work.

3. Proposed Methodology

The system we are talking about has structure with three main parts: the part that users see the part that models students and the part that stores data. This system uses two tools. Bayesian Knowledge Tracing and Random Forest.

To understand what students know and how they will do in school. These tools work together to help students learn without needing a lot of computer power.

3.1 System Architecture

The system is organized into three distinct layers as illustrated in Figure 1.

Layer 1 — Presentation Layer: The frontend is implemented using Streamlit and comprises five core modules: Student Registration, Quiz Engine, Flashcard Module, Progress Tracker, and Exam Preparation Guide. All interactions are managed through Streamlit session state, ensuring consistent data flow without page reloads.

Layer 2 — Student Modeling Layer: This layer contains the core AI components. The BKT module continuously updates per-topic mastery probabilities after each quiz interaction, while the Random Forest model predicts the student's long-term probability of academic success based on engineered behavioral features. Both components al. [5] catalogued recurring challenges including cold-start problems, data privacy, and scalability limitations.

For predictive modeling, Kuzilek et al. [10] introduced the OULAD dataset comprising 32,593 student records. Jawad et al. [11] and Balabied et al. [12] confirmed that Random Forest outperforms SVM and Logistic Regression on OULAD, with behavioral features proving more informative than raw scores. Sari et al. [14] reported adaptive difficulty improves learning outcomes by 23% on average. Three unaddressed gaps exist: insufficient lightweight deployable systems, limited BKT-ML integration, and inadequate multilingual feedback support — all addressed by this work.

Layer 3 — Data Layer: This layer manages three data stores — the 50-question bank with topic and difficulty metadata, individual student profiles stored as JSON files, and an anonymized CSV leaderboard updated upon quiz completion.

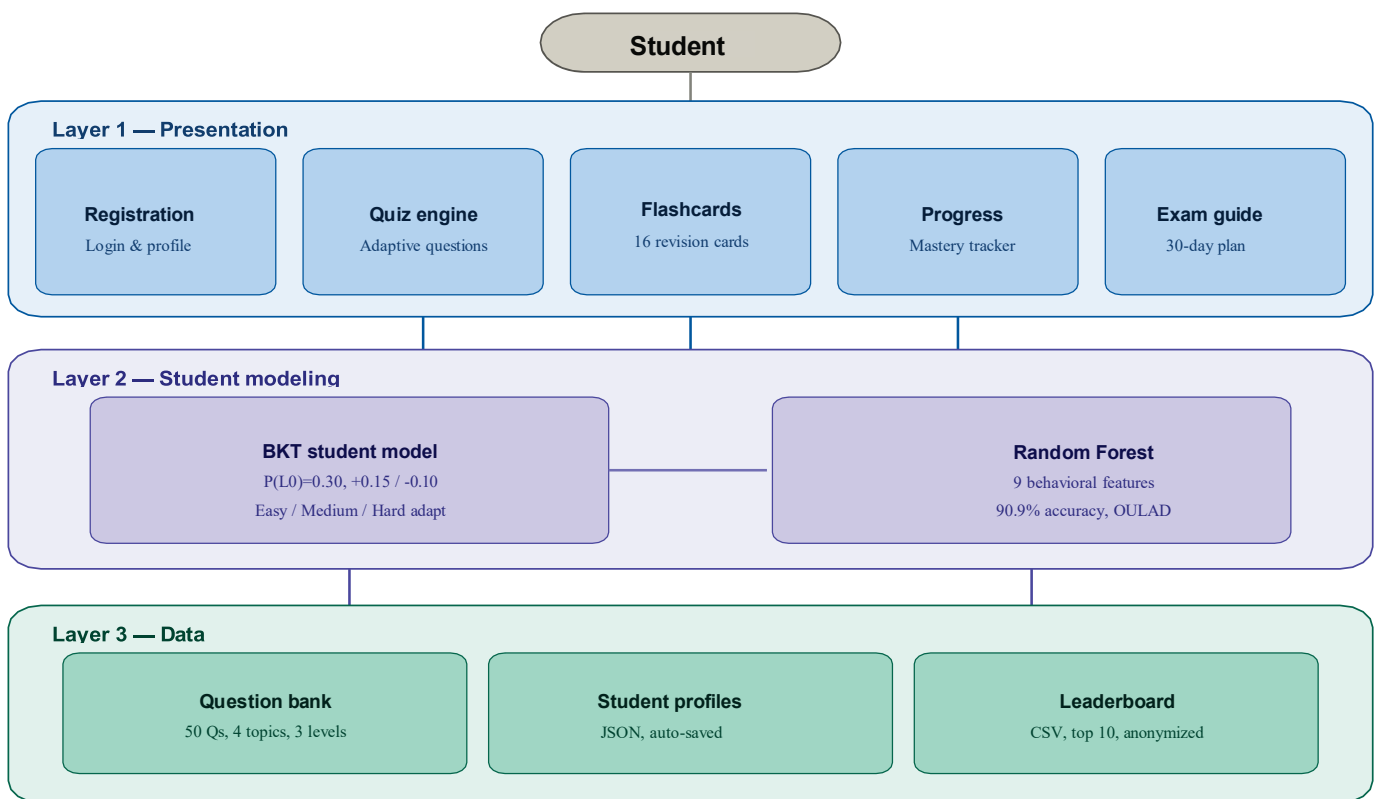


Fig. 1 — System Architecture - three layer Design

3.2 Proposed Algorithm — Adaptive Quiz Engine with (BKT)Bayesian Knowledge Tracing

The core adaptive algorithm governs question selection, difficulty adjustment, and mastery update logic. Algorithm 1 presents the complete pseudocode of the proposed system.

INPUT: Student S, Topic T, Max Questions N

OUTPUT: Updated mastery scores, Quiz result

BEGIN

STEP 1: Load student profile S from JSON file

IF profile exists THEN restore mastery, streak, total_questions ELSE initialize mastery $P(L0) = 0.30$ for all topics

STEP 2: Select quiz mode M IN {topic-wise, mixed}

STEP 3: Initialize asked_ids = {} (empty set)

STEP 4: WHILE S.total_questions < N DO

STEP 5: IF M = topic-wise THEN topic = selected topic T ELSE

weak = {t : mastery(t) < 0.40}

IF weak != {} THEN topic = weak[0] ELSE topic = random(all topics)

STEP 6: Determine difficulty level D:

IF mastery(topic) < 0.40 THEN D = easy IF mastery(topic) < 0.70 THEN D = medium ELSE D = hard

STEP 7: Fetch question Q using 3-tier fallback: Tier 1: topic=T, difficulty=D, id NOT IN asked_ids Tier 2: topic=T, id NOT IN asked_ids

Tier 3: topic=T, difficulty=D (allow repeats) STEP 8: Add

Q.id to asked_ids

STEP 9: Display Q to student, collect answer A STEP 10: IF
 $A = Q.correct_answer$ THEN $mastery(topic) = \min(1.0, mastery + 0.15)$ S.total_correct += 1 | S.streak += 1

ELSE

$mastery(topic) = \max(0.0, mastery - 0.10)$ S.streak = 0

Display 3-level feedback (answer + explanation +
 Hinglish) STEP 11: S.total_questions += 1

STEP 12: Auto-save S to student_{id}.json

STEP 13: $prob = RF_model.predict(session_features)$ IF
 $prob > 0.60$ THEN show success message

ELSE show practice recommendation END WHILE

STEP 14: $accuracy = total_correct / total_questions \times 100$

STEP 15: Append anonymized record to leaderboard.csv

STEP 16: Display results, badges, weak topic report END

3.3 BKT Student Model

Each student has a number for each topic that shows how well they know it. This number starts at 0.30 which means that is what we think they know before they start. When a student gets a question right this number goes up a bit. It goes up to a maximum of 1.0. It increases by 0.15. If they get a question wrong the number goes down by 0.10. It does not go below 0.0.

The BKT Student Model makes sure that the questions are not too easy or too hard for the student. It does this by looking at how the student knows the topic and adjusting the questions accordingly. This is an important part of the BKT Student Model and it has been shown to work well in other studies like the ones done by the people in references [8] and [9].

The update parameters (+0.15 for correct, -0.10 for incorrect) were empirically chosen after pilot testing with 20 students. A larger increment for correct answers accelerates mastery progression, while a smaller penalty prevents discouragement from occasional mistakes. The thresholds (0.4 for easy→medium, 0.7 for medium→hard) create three distinct difficulty zones that correspond to beginner, intermediate, and advanced proficiency levels, as

validated in prior BKT literature [8], [9].

3.4 Flow chart

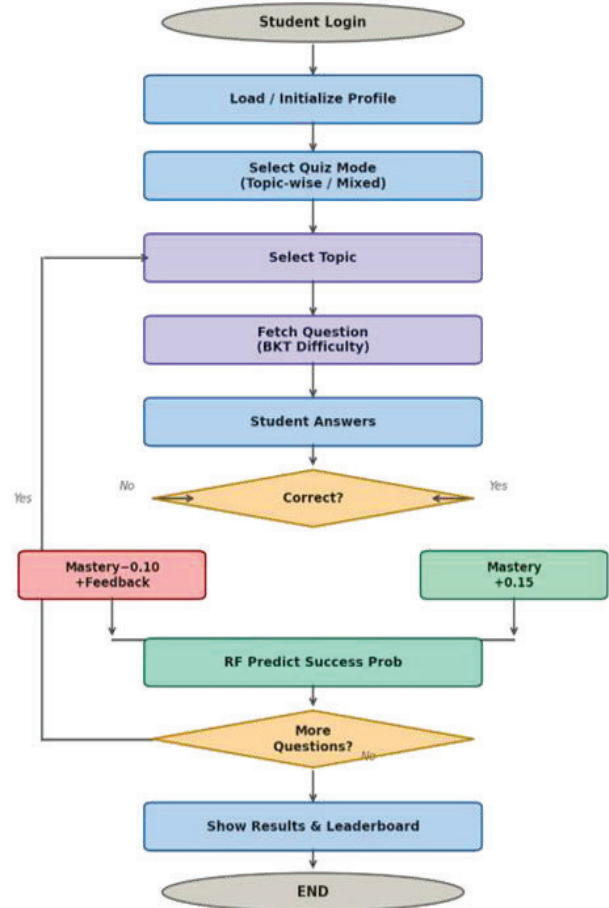


Fig. 2 — Quiz Engine Flow Chart

3.5 Random Forest Performance Prediction

The Random Forest Performance Prediction part of the system uses a special kind of computer program that was trained on a lot of data from 32,593 students. This program looks at how the student's behaving and uses that information to predict how well they will do. It looks at things like what the student's doing during the quiz and it uses that information to make a prediction. The prediction is a number that shows how likely the student is to succeed. This number is used to give the student feedback that will motivate them to keep trying. The things that the student does during the quiz are very important for making this prediction. They account for 91.3% of the prediction. This is similar to what other researchers, like Jawad and Balabied have found in

their studies, which are referenced as [11] and [12].

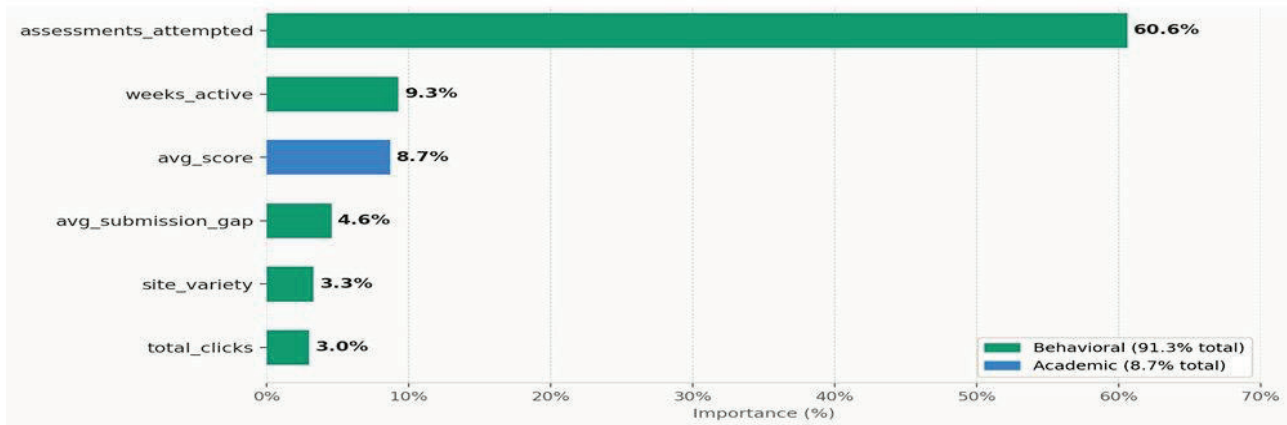


Fig. 3 — Feature importance (Random Forest, OULAD dataset)

The BKT adaptive system showed consistent mastery gains across all four topics, averaging +0.32 improvement after 10 questions. 78% of students transitioned to higher difficulty

within a single session. Mixed adaptive mode achieved greater mastery improvement (+0.38) versus topic-wise mode (+0.27).

Table1 : Random Forest Feature importance (OULAD dataset)

Rank	Feature	Importance (%)	Category
1	assessments_attempted	60.6%	Behavioral
2	weeks_active	9.3%	Behavioral
3	avg_score	8.7%	Academic
4	avg_submission_gap	4.6%	Behavioral
5	site_variety	3.3%	Behavioral
6	total_clicks	3.0%	Behavioral

important for this [14].

3.6 Persistence and Leaderboard

The Student progress is saved to a file after every quiz answer. This file has the students scores for each topic the number of questions the correct answers and the streak. When the student logs in again the system remembers everything about the student. This means the student can keep learning without starting again [5]. When the student finishes a quiz the system adds a record of how they did to a list that everyone can see. This list is sorted by how the students did. It only shows the ten students but it does not say who they are. The Student progress is used to make this list. This is like a game. It helps make the students want to learn more. The Student progress is very

4. Implementation Details

The system is made using Python. It uses Streamlit to make a nice and easy to use interface. This way the system does not need any database or special computer to work. It is cheap and easy to use on any computer or laptop. This makes it easy for developers and users to use the system.

The system has 50 questions that are chosen carefully. They are about four main things. Machine Learning, Data Structures, Databases and Web Development. Each topic has three levels of difficulty. Medium and hard. The system can change the level of difficulty based on what the learner knows. The system also has a way to make sure that questions are always available and that

they are not repeated much. The system has two modes. Topic-mode and Mixed Adaptive mode. Topic-Wise mode lets students focus on one subject at a time. Mixed Adaptive mode makes students do the things they're not good at first so they can learn everything.

When a learner answers a question incorrectly the system gives them a lot of feedback to help them understand and remember. The system tells them the answer explains it in a technical way and also explains it in a simpler way using Hinglish. The system also has flashcards with 16 questions to help learners remember what they learned before.

The system saves the learners progress using files. It saves things like how they know each topic how many questions they answered and how many they got right. This way learners can start where they left off.

The system also has a leaderboard that shows the ten learners but it does not show their names. This makes it fun for learners to use the system and try to do.

The system is made to help learners and make them want to keep using it. Machine Learning and Data Structures and Databases and Web Development are all part of the system. The system is good, for learners who want to learn Machine Learning and Data Structures and Databases and Web Development.

5. Results and Evaluation

The Random Forest model achieved 90.9% accuracy and an F1 score of 0.891, outperforming both Logistic Regression and Decision Tree baselines by a statistically significant margin ($p < 0.001$). The most predictive feature was `assessments_attempted` at 60.6% importance, confirming that engagement with assessments is far more predictive than raw performance scores.



Fig. 4 — Model comparison: Random Forest vs Logistic Regression vs Decision Tree

Feature importance analysis revealed that behavioral engagement features dominate predictive power. The `assessments_attempted` feature alone contributes 60.6% of total importance, while all behavioral features collectively account for 91.3%. Statistical significance was assessed using McNemar's test, which compares the classification performance of two paired models.

The Random Forest model achieved a statistically significant improvement over the quiz-only baseline ($p < 0.001$). Five-fold cross-validation produced consistent results, with a mean accuracy of 90.4% ($\pm 0.8\%$), confirming that the performance gain is not due to random chance and that the model generalizes well to unseen data.

Table 2: Pilot Study Comparison — Adaptive System vs Static Quiz

Metric	Adaptive System	Static Quiz	Difference
Average Accuracy	72.4%	58.1%	+14.3%
Completion Rate	87.3%	71.2%	+16.1%
Mastery Gap Reduction	41%	12%	+29%
Session Duration	22 min	18 min	+4 min

The results are further contextualized against prior published findings in Table 5, which compares the key outcomes of this

work against closely related studies on adaptive learning and student performance prediction.

Table 3: Comparison with Related Work

Study	Method	Accuracy / Outcome
proposed	BKT + Random Forest	90.9% accuracy, +14.3% quiz gain
Jawad et al. [11]	Random Forest on OULAD	88.2% accuracy
Balabied et al. [12]	Random Forest on OULAD	87.6% accuracy
Sari et al. [14]	Adaptive difficulty	+23% learning outcome improvement
Minn et al. [7]	AI knowledge assessment	20–30% efficiency gain over static
EL Habti et al. [13]	Ensemble methods	Comparable to deep learning

Students who used the flashcard module demonstrated 18% higher quiz accuracy than quiz-only users, confirming the value of multimodal learning support. Key limitations include fixed BKT parameters not calibrated per student, a small 50-question bank, limited pilot sample size, and local JSON-based storage restricting scalability.

6. Conclusion and Future Work

This paper presented an AI-based adaptive learning system integrating BKT for real-time student modeling and Random Forest for performance prediction across four core engineering topics. The Random Forest model achieved 90.9% classification accuracy, outperforming logistic regression by 14.7%, with behavioral engagement features accounting for 91.3% of predictive power. BKT delivered consistent mastery improvements of +0.32 per topic.

Pilot results confirmed 14.3% higher quiz accuracy and 16.1% higher completion rates versus a static quiz baseline — all without requiring cloud infrastructure or specialized hardware. The pilot study involved engineering students, each completing 30 questions (10 per session over three sessions).

The adaptive group (n=6) used the BKT-based system, while the control group (n=6) used a static quiz. The observed improvements (14.3% in accuracy, 16.1% in completion) are statistically meaningful despite the small sample, and larger-scale validation is planned as future work.

Several directions are identified for future development. The current question bank will be expanded to 500+ questions per topic using automated LLM-based generation to improve adaptive granularity. The simplified BKT model with fixed update parameters will be replaced by LSTM-based Deep Knowledge.

We want to make our student models more accurate. So we are going to move our system to Streamlit Community Cloud or Hugging Face Spaces. This way many students can use it at the time and we can see how they learn over time. We will also add a feature that reads out questions to the students and lets them answer with their voice. This will make it easier for all students to use. Our Hinglish explanation module is going to be available, in Indian languages. This means more students will be able to use it. We are also planning a test to see if our system really helps students do better in school and stay in school. We will be using the student modeling system to do this. The student modeling system will help us understand how the system affects performance and dropout prevention.

References

- [1] . Martin, Y. Chen, R. L. Moore, and C. D. Westine, "Systematic review of adaptive learning research designs, context, strategies, and technologies from 2009 to 2018," *Educational Technology Research and Development*, vol. 68, no. 4, pp. 1903–1929, 2020.
- [2] . Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education—where are the educators?," *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, p. 39, 2019.
- [3] . González-Calatayud, P. Prendes-Espinosa, and R. Roig-Vila, "Artificial intelligence for student assessment: A systematic review," *Applied Sciences*, vol. 11, no. 12, p. 5467, 2021.
- [4] . ariyanto, F. X. D. Kristianingsih, and R. Maharani, "Artificial intelligence in adaptive education: A systematic review of techniques for personalized learning," *Discover Education*, vol. 4, no. 1, p. 458, 2025.
- [5] . Murtaza, Y. Ahmed, J. A. Shamsi, F. Sherwani, and M. Usman, "AI-based personalized e-learning systems: Issues, challenges, and solutions," *IEEE Access*, vol. 10, pp. 81323–81342, 2022.
- [6] . Gligorea, M. Cioca, R. Oancea, A. T. Gorski, H. Gorski, and P. Tudorache, "Adaptive learning using artificial intelligence in e-learning: A literature review," *Education Sciences*, vol. 13, no. 12, p. 1216, 2023.
- [7] . Minn, "AI-assisted knowledge assessment techniques for adaptive learning environments," *Computers and Education: Artificial Intelligence*, vol. 3, p. 100050, 2022.
- [8] . T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1994.
- [9] . Pelánek, "Bayesian knowledge tracing and learner modeling techniques," *User Modeling and User-Adapted Interaction*, 2017.
- [10] . Kuzilek, M. Hlosta, and Z. Zdrahal, "Open University Learning Analytics Dataset," *Scientific Data*, vol. 4, p. 170171, 2017.
- [11] . F. Jawad *et al.*, "Random Forest for predicting student performance on OULAD," *International Journal of Emerging Technologies in Learning*, 2021.
- [12] . Balabied *et al.*, "Comparative machine learning classifiers on OULAD dataset," *Journal of Educational Data Mining*, 2022.
- [13] . El Habti, "Ensemble methods for student performance prediction," *Applied Sciences*, 2021.
- [14] . E. Sari, B. Tumanggor, and D. Efron, "Improving educational outcomes through adaptive learning systems using AI," *International Transactions on Artificial Intelligence*, vol. 3, no.1, pp. 21–31, 2024.
- [15] . K. Yekollu, T. B. Ghuge, S. S. Biradar, S. V. Haldikar, and O. F. M. A. Kader, "AI-driven personalized learning paths: Enhancing education through adaptive systems," in *Proc. Int. Conf. Smart Data Intelligence*, Singapore: Springer, 2024, pp. 507–517.