TeachMate AI: An Integrated AI-Powered Grading System and Teacher Guidance Automation for O/L ICT Education in Sri Lanka

Pathiraja P.U.M., **Wanniarachchi W.A.P.M.**, **Hettiarachchi R.H.**, **Jayasooriya L.T.**
Faculty of Computing, Sri Lanka Institute of Information Technology (SLIIT), Malabe, Sri Lanka
Supervisors: **Samantha Rajapaksha** (samantha.r@sliit.lk), **Bhagyanie Chathurika** (bhagyanie.c@sliit.lk)

Abstract

The evolving landscape of Information and Communication Technology (ICT) education in Sri Lanka, particularly with recent curriculum reforms emphasizing competency-based learning and AI integration for Grades 10–11, demands innovative assessment and personalized learning approaches. Traditional manual grading and generic teaching methods fail to provide timely feedback or adaptive learning paths, creating significant barriers to effective education delivery. This research introduces TeachMate AI, an integrated AI-driven Learning Management System (LMS) comprising four intelligent components: (1) an automated answer sheet grading system using a Vision-Language Model (VLM) — Qwen2-VL-7B-Instruct — combined with Sentence-BERT semantic similarity; (2) a knowledge graph-based reteaching guidance module powered by Bayesian Knowledge Tracing (BKT) and a custom Graph Neural Network (GNN); (3) a Bloom's Taxonomy-aligned question paper generator using DistilBERT and Integer Linear Programming; and (4) a personalized learning plan generator with adaptive resource tracking via Logistic Regression. By combining Optical Character Recognition (OCR), Natural Language Processing (NLP), machine learning, and knowledge graph technologies, the system automates the complete teacher–student feedback loop. Experimental evaluations with 120 Grade 11 students over one academic term demonstrate an 84.9% reduction in grading time, 0.83 F1-score in learning gap prediction, 86% accuracy in Bloom's classification, and a statistically significant 14.4% improvement in student learning outcomes ($p < 0.001$).

Keywords: Learning Management System; Automated Grading; VLM; Bayesian Knowledge Tracing; Graph Neural Networks; Bloom's Taxonomy; Personalized Learning; O/L ICT; Sri Lanka Education

1. INTRODUCTION

The Sri Lankan education system is undergoing transformative reforms, particularly in ICT at the Ordinary Level (O/L). Recent curriculum updates have transitioned O/L ICT from an optional subject to a mandatory competency-based programmer, emphasizing hands-on projects and introducing artificial intelligence fundamentals. From 2026, the Department of Examinations will conduct reformed O/L examinations, including Information Technology as one of seven core subjects, shifting from traditional pass/fail grading to a Grade Point Average (GPA) system.

These reforms expose critical deficiencies in the current teaching ecosystem. Teachers face overwhelming workloads — manually grading answer sheets for large classrooms, struggling to identify individual learning gaps, and lacking tools for timely personalized interventions [1]. Students receive delayed feedback and generic instruction that fails to address their unique conceptual weaknesses. Existing Learning Management Systems (LMS) primarily focus on content delivery and basic grade recording without integrating intelligent assessment automation, semantic analysis of student understanding, or personalized learning-path generation [2].

Commercial platforms offering automated grading are generally limited to multiple-choice questions and cannot evaluate subjective written responses requiring semantic comprehension [3]. Furthermore, most systems cannot construct detailed knowledge graphs representing student mastery or generate adaptive reteaching recommendations based on identified gaps.

This research addresses these limitations by introducing TeachMate AI, an integrated AI-driven LMS specifically designed for Sri Lankan O/L ICT education. The system comprises four synergistic components:

- AI-Based Answer Sheet Grading System: Automates the extraction and evaluation of handwritten responses using VLM-based OCR and NLP, providing marks and constructive feedback.
- Reteaching Guidance & Knowledge Graphs: Analyses graded responses to build dynamic student knowledge graphs using BKT and GNN inference, generating personalized reteaching recommendations.
- Automated Question Paper Generation: Creates Bloom's Taxonomy-aligned question papers using DistilBERT classification and Integer Linear Programming optimization.
- Personalized Learning Plan Generator: Synthesizes performance data to produce individualized plans with adaptive resource tracking, accessible through a mobile interface.

The primary objectives are to: (1) significantly reduce teacher workload in assessment and grading; (2) enable precise identification of concept-level learning gaps; (3) automate generation of pedagogically sound assessments; and (4) provide students with personalized, actionable learning guidance through an end-to-end intelligent feedback loop.

2. LITERATURE REVIEW

A. Automated Assessment and Grading

Traditional assessment methods are time-consuming and prone to human error and bias. Early automation focused primarily on multiple-choice questions [4]. Recent advances in NLP have enabled sophisticated approaches; Ramesh and Sanampudi [5] demonstrated that transformer-based models such as BERT can achieve accuracy comparable to human graders for short written answers. Sung et al. [6] applied semantic similarity using sentence embeddings to score essays, capturing conceptual understanding beyond keyword matching. OCR technology has evolved significantly, with modern systems achieving high accuracy on printed text [7]. Hybrid approaches combining traditional OCR with deep learning post-correction substantially improve recognition accuracy for handwritten academic documents [8]. Vision-Language Models (VLMs) represent the frontier, enabling simultaneous perception of layout and text for complex handwritten documents.

B. Knowledge Representation and Learning Analytics

Knowledge graphs have emerged as powerful tools for reasoning student learning states. Chen et al. [9] introduced a framework using knowledge graphs to model prerequisite relationships between concepts. Bayesian Knowledge Tracing (BKT), proposed by Corbett and Anderson [10], provides a probabilistic approach to estimating student mastery over time using four parameters: prior knowledge probability (p_init), learning probability (p_learn), slip probability (p_slip), and guess probability (p_guess). More recently, Graph Neural Networks (GNNs) have been applied to knowledge tracing; Nakagawa et al. [11] demonstrated that GNN-based models outperform traditional BKT by capturing complex concept dependencies through message-passing operations.

C. Adaptive Learning and Personalization

Personalized learning systems tailor educational content and pacing to individual student needs [13]. Recommender systems adapted from e-commerce have been successfully applied to education, with collaborative filtering [14] and content-based hybrid approaches [15] improving recommendation relevance. Mobile learning applications significantly improve student motivation and flexibility [16]. The ResourceCompletionPredictor pattern — using logistic regression on resource-type embeddings and interaction history — represents an emerging approach to adaptive resource sequencing.

D. Bloom's Taxonomy in Assessment Design

Bloom's Taxonomy remains the dominant framework for designing assessments with appropriate cognitive complexity [17]. Automated classification has been explored using SVMs [18] and pre-trained language models; Goel and Singh [19] fine-tuned DistilBERT for Bloom's classification, achieving over 90% accuracy. Our work extends this by combining cosine similarity over Bloom-level indicator embeddings with keyword-boosting heuristics.

E. Research Gaps

While individual components — automated grading, knowledge graphs, question generation, and personalized learning — have been studied separately, few systems integrate all elements into a cohesive platform. Most existing research focuses on Western contexts with well-resourced digital infrastructures, underexploring challenges specific to developing countries [20]. Furthermore, most automated grading systems stop at assigning marks without actionable reteaching recommendations [21]. This research addresses these gaps by developing an integrated, context-aware AI-driven LMS encompassing the complete teaching-learning-assessment cycle.

3. METHODOLOGY

A. System Architecture Overview

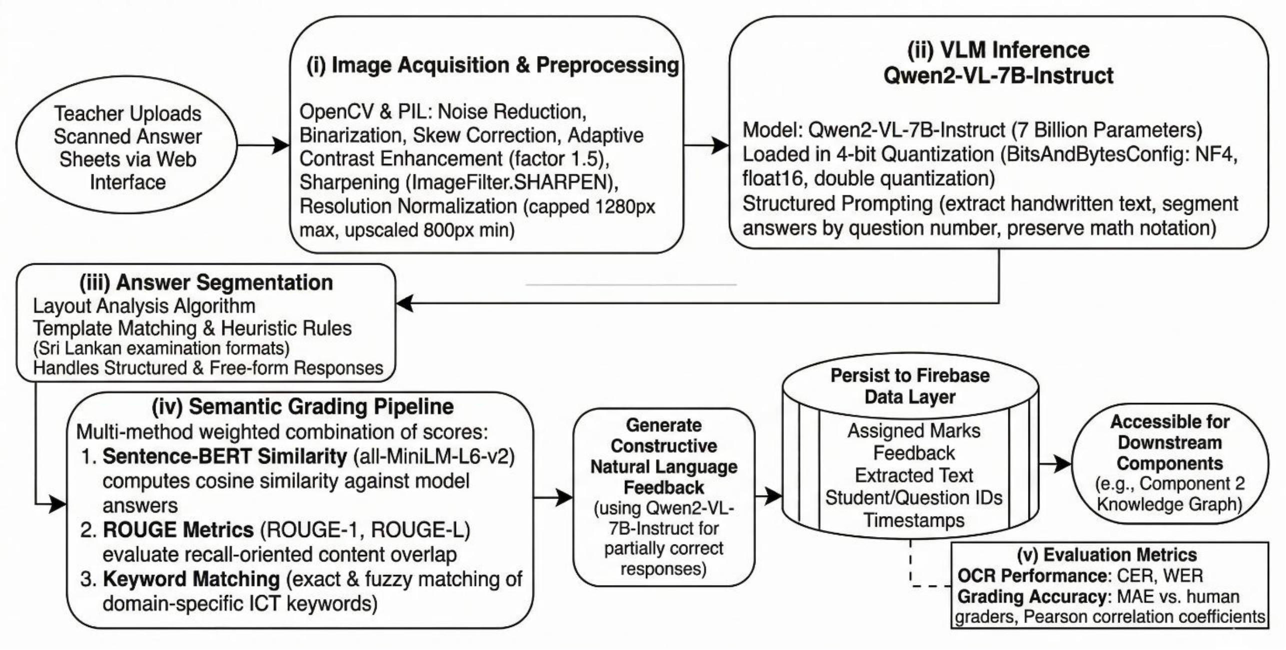
The proposed AI-Driven LMS integrates four interdependent components designed to work both independently and as a unified ecosystem. The architecture follows a modular design with shared data models and APIs:

- Frontend Layer: React.js web interface for teachers; React Native mobile app for students.
- Backend Layer: RESTful APIs implemented using Flask and FastAPI with dedicated services per component.
- Data Layer: Firebase Realtime Database for shared persistence and real-time synchronization.
- AI/ML Layer: Specialized models for OCR, NLP, knowledge graph construction, and adaptive recommendation generation.

All components communicate through a centralized API gateway, handling authentication, data routing, and inter-component message passing. Outputs from one module (e.g., graded answers from Component 1) are directly consumed by downstream modules (e.g., Knowledge Graph in Component 2), enabling a continuous feedback loop.

B. Component 1: VLM-Based OCR Engine and AI Grading System

This component automates the evaluation of handwritten O/L ICT answer sheets. Figure 1 illustrates the individual workflow for Component 1.



[Figure 1: Component 1 — OCR and Grading Pipeline Flow]

The workflow proceeds through the following stages:

(i) Image Acquisition and Preprocessing:

Teachers upload scanned answer sheet images through the web interface. Each image undergoes preprocessing via OpenCV and PIL: noise reduction, binarization, skew correction, adaptive contrast enhancement (factor 1.5), sharpening (ImageFilter.SHARPEN), and resolution normalization (capped at 1280px on the longest side, upscaled to 800px minimum) to optimize VLM input quality.

(ii) Vision-Language Model (VLM) Inference — Qwen2-VL-7B-Instruct:

The system employs Qwen2-VL-7B-Instruct (7 billion parameters) as the primary OCR engine, loaded in 4-bit quantization using BitsAndBytesConfig (NF4 quantization type, float16 compute dtype, double quantization enabled) to fit within GPU memory constraints. The VLM is prompted with a structured instruction to extract handwritten text, segment answers by question number, and preserve mathematical notation. 4-bit NF4 quantization reduces GPU memory from approximately 14GB to under 5GB while maintaining over 95% of full-precision accuracy on text extraction tasks.

(iii) Answer Segmentation:

After text extraction, a layout analysis algorithm divides recognised text into individual answers based on detected question numbers and visual markers. The process incorporates template matching and heuristic rules tailored to standard Sri Lankan examination answer sheet formats, handling both structured (question-numbered) and free-form responses.

(iv) Semantic Grading Pipeline:

Each segmented answer undergoes multi-method semantic evaluation against the corresponding marking scheme. Three complementary techniques are combined:

- Sentence-BERT Similarity: The all-MiniLM-L6-v2 model computes cosine similarity scores between student response embeddings and model answer embeddings, capturing semantic equivalence beyond lexical overlap.
- ROUGE Metrics: ROUGE-1 and ROUGE-L scores evaluate recall-oriented content overlap to ensure key factual points are present.
- Keyword Matching: Domain-specific ICT keywords defined in the marking scheme are detected using exact and fuzzy matching to verify technical terminology.

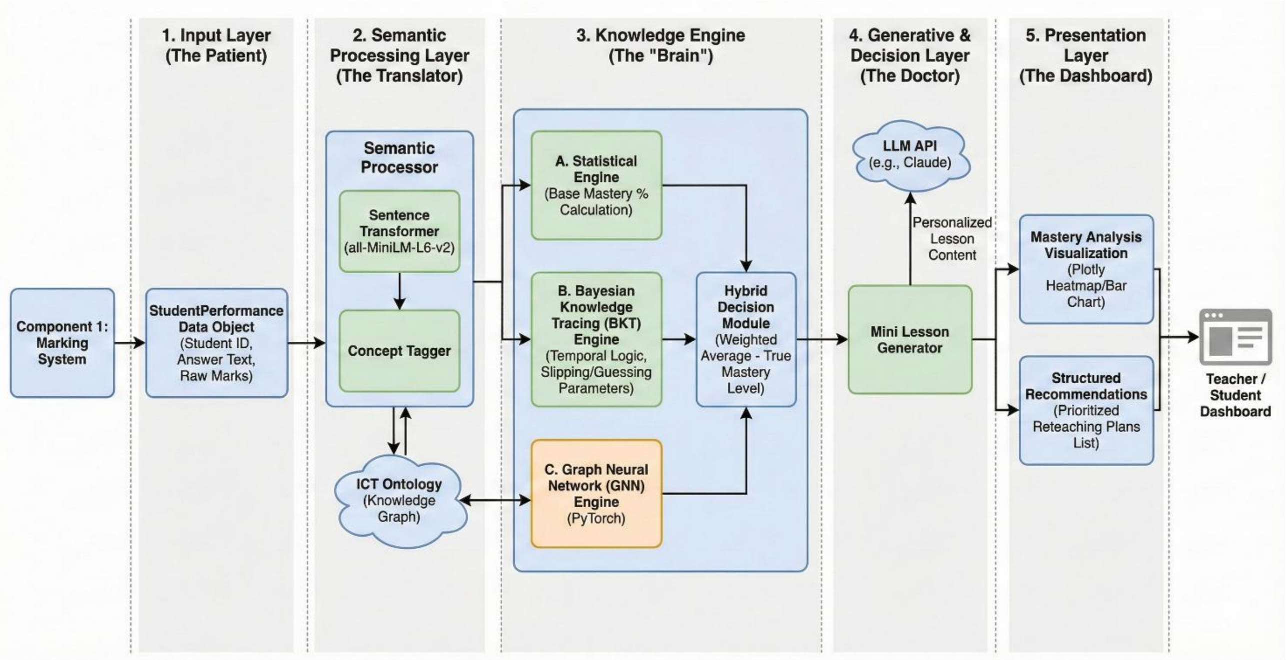
Final marks are assigned proportionally based on a weighted combination of these three scores. For partially correct responses, the Qwen2-VL-7B-Instruct model generates constructive natural language feedback highlighting missing concepts and potential errors. All outputs — assigned marks, feedback, extracted text, student/question identifiers, and timestamps persisted to Firebase for downstream component access.

(v) Evaluation Metrics:

OCR performance is measured using Character Error Rate (CER) and Word Error Rate (WER) on a manually annotated test set. Grading accuracy is validated by comparing system-generated scores against three independent human graders using Mean Absolute Error (MAE) and Pearson correlation coefficients.

C. Component 2: Knowledge Graphs, BKT, and Reteaching Guidance System

This component extends automated grading outputs to construct detailed models of student understanding and generate targeted reteaching strategies. Figure 2 illustrates the workflow for Component 2.



[Figure 2: Component 2 — Knowledge Graph Construction and BKT/GNN Inference Flow]

(i) ICT Concept Ontology:

The system employs a rich domain ontology constructed from the official Grade 10–11 O/L ICT syllabus, covering over 150 distinct concept nodes organised by term and lesson. Each ontology node contains: lesson_id (e.g., 10.1 for Grade 10 Term 1 Lesson 1), parent concept (enabling prerequisite modelling), child concepts (for hierarchical traversal), domain keywords, and lesson descriptions. Semantic relationships include prerequisite dependencies (e.g., 'binary_conversion' → 'number_systems'), part-of hierarchies (e.g., 'cpu' ⊂ 'computer_components'), and related-to associations (e.g., 'logic_gates' ~ 'boolean_expressions').

(ii) NLP-Based Concept Extraction:

A BERT-based Named Entity Recognition (NER) model, fine-tuned on O/L ICT content, detects domain-specific entities in student answers (Technical Terms, Process Names, Tool/Software). Extracted entities are linked to corresponding ontology nodes via keyword matching and semantic similarity. The NER model achieved a precision of 0.91, a recall of 0.87, and an F1-score of 0.89 on the manually annotated test set.

(iii) Student Knowledge Graph (Neo4j):

A personalized knowledge graph is generated and continuously updated in Neo4j for each student. The graph structure includes: a central Student node connected to ConceptMastery nodes (each annotated with mastery_score ∈ [0,1], attempt_count, last_updated), and Assessment edges linking concepts to specific test instances (with performance_score and temporal information).

(iv) Bayesian Knowledge Tracing (BKT):

BKT models the probability that a student has mastered each concept based on historical performance. The system implements the four-parameter BKT model with the following parameter initialization based on empirical calibration on the O/L ICT dataset:

BKT Parameter	Value	Description
p_init	0.50	Prior probability of concept mastery
p_learn	0.30	Probability of learning per attempt
p_forget	0.05	Probability of forgetting a mastered concept
p_slip	0.10	Error probability despite mastery
p_guess	0.20	Correct answer probability without mastery

Table 1. BKT Parameter Configuration for O/L ICT Domain

Parameters are optimized using the Expectation-Maximization (EM) algorithm trained on historical student performance data. The BKT update equations propagate mastery probabilities forward in time, enabling real-time identification of concepts requiring reinforcement.

(v) Graph Neural Network (GNN) Inference:

A custom Graph Attention Network (GAT) architecture is applied to the student knowledge graph to predict latent weaknesses and prerequisite gaps by identifying correlated learning deficiencies across the concept graph. The GNN architecture comprises three message-passing layers:

- Layer 1 (W1: input_dim → 64): Initial neighborhood aggregation with ReLU activation.
- Layer 2 (W2: 64 → 64): Deep feature refinement with 0.3 dropout regularization.
- Layer 3 (W3: 64 → 64): High-order neighborhood aggregation.
- Attention Mechanism: Linear attention weights (64 → 1) applied during aggregation to weight neighbor contributions by relevance.
- Output Layer (64 → 1): Sigmoid-activated mastery prediction per concept node.

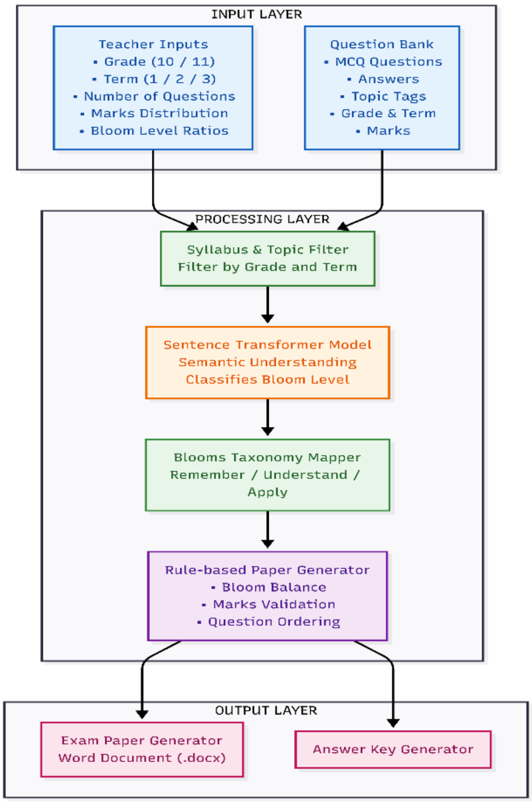
The message-passing operation aggregates neighbor features via adjacency matrix multiplication: $h'(v) = \sigma(W \cdot (h(v) + \Sigma\_u \in N(v)\ \alpha\_vu \cdot h(u)))$, where α_vu are learned attention coefficients. The GNN combined with BKT achieved an F1-score of 0.83, outperforming rule-based (0.61), collaborative filtering (0.69), and BKT-only (0.76) baselines.

(vi) Recommendation Engine and Teacher Dashboard:

Based on BKT and GNN outputs, a recommendation engine generates prioritized reteaching strategies: identifying target concepts ranked by downstream learning impact and mapping appropriate instructional resources (mini-lessons, practice questions, video tutorials). Sequential learning paths respect prerequisite relationships from the ontology. An interactive teacher dashboard visualizes class-wide mastery heatmaps, individual student knowledge graphs, and AI-generated intervention plans. Teachers retain full control to approve, modify, or reject suggestions. A continuous feedback loop updates model parameters based on teacher decisions and post-intervention assessment outcomes.

D. Component 3: Bloom's Taxonomy-Aligned Question Paper Generator

This component generates pedagogically sound question papers from a teacher-curated question bank. Figure 3 illustrates the Component 3 workflow.



[Figure 3: Component 3 — Question Classification and Paper Generation Flow]

(i) Question Bank Management:

Teachers manage a question bank through the web interface, providing for each question: question text, model answer, difficulty level (Easy/Medium/Hard), associated O/L ICT topic, and maximum marks.

(ii) Bloom's Taxonomy Classification (DistilBERT + Semantic Similarity):

Each question is automatically classified into one of six Bloom's cognitive levels — Remember, Understand, Apply, Analyse, Evaluate, and Create — using a dual-method approach. First, the all-MiniLM-L6-v2 model encodes the question text and computes cosine similarities against pre-encoded Bloom's level indicator phrase embeddings. Second, a keyword-boosting heuristic adjusts scores based on the presence of strong action verb indicators (e.g., 'define' → Remember, 'compare' → Understand, 'apply' → Apply, 'analyse' → Analyse, 'evaluate' → Evaluate, 'design' → Create). The final classification uses the highest-scoring level after boost adjustment. A fine-tuned DistilBERT model trained on annotated O/L ICT questions provides the primary classification signal for production use.

(iii) Constrained Paper Generation (Integer Linear Programming):

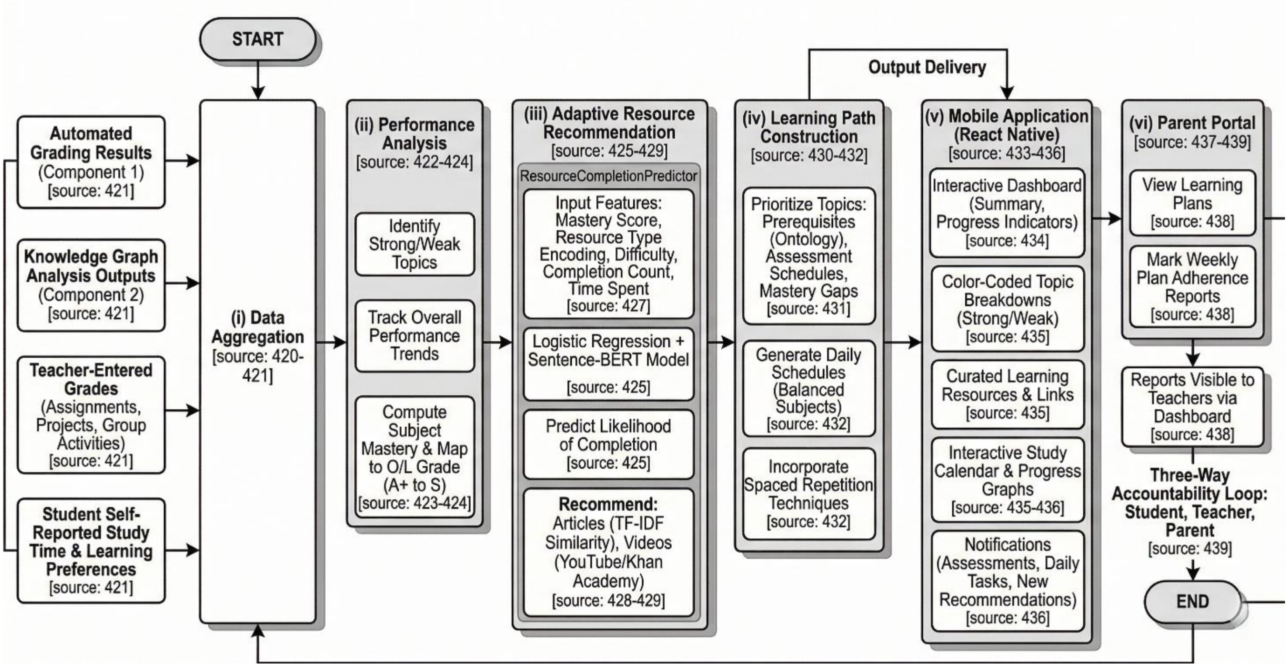
Teachers specify paper generation criteria: total marks, question count, desired Bloom's level distribution, topic coverage constraints, and difficulty balance. An Integer Linear Programming (ILP) optimisation formulates question selection as a constraint satisfaction problem with an objective function maximising topic diversity and minimising repetition across recent assessments. For large question banks, a greedy heuristic pre-selects high-impact questions before applying ILP.

(iv) Output and Version Control:

The system assembles questions ordered by ascending Bloom's cognitive complexity and generates: a student-facing PDF question paper, a teacher-facing marking scheme with detailed answers and mark allocations, and metadata (question IDs, Bloom's levels) stored in Firebase. Version control prevents excessive repetition across consecutive assessments by tracking previously generated paper configurations.

E. Component 4: Personalized Learning Plan Generator with Adaptive Resource Tracking

This component synthesizes performance data from multiple sources to generate individualized student learning plans delivered via a React Native mobile application. Figure 4 illustrates the Component 4 workflow.



[Figure 4: Component 4 — Personalized Learning Plan Generation Flow]

(i) Data Aggregation:

The system aggregates data from four sources: automated grading results (Component 1), knowledge graph analysis outputs (Component 2), teacher-entered grades for assignments, projects, and group activities, and student self-reported study time and learning preferences.

(ii) Performance Analysis:

Machine learning models process aggregated data to identify strong topics (consistently high performance), weak topics (low marks and detected knowledge gaps), and overall performance trends (improvement or decline over time). The system computes a subject mastery level and maps it to a grade (A+, A, B+, B, C+, C, S) aligned with Sri Lanka's O/L grading system.

(iii) Adaptive Resource Recommendation (Logistic Regression + Sentence-BERT):

The ResourceCompletionPredictor employs a Logistic Regression model trained on student resource interaction histories to predict the likelihood of resource completion. Input features include: concept mastery score, resource type encoding (article/video/interactive/practice via LabelEncoder), difficulty level, previous completion count, and average time spent on similar resource types. Article recommendations use TF-IDF similarity against a curated content database; video recommendations link to YouTube and Khan Academy tutorials aligned with identified gaps.

(iv) Learning Path Construction:

A personalized learning path algorithm prioritizes topics according to prerequisite relationships from the ontology, upcoming assessment schedules, and mastery gaps. The generated study plan proposes daily schedules balancing multiple subjects and incorporates spaced repetition techniques for long-term knowledge retention.

(v) Mobile Application Features:

The React Native mobile app delivers: an interactive dashboard with overall performance summary and subject-wise progress indicators; color-coded topic breakdowns (strong/weak); curated learning resources with direct access links; an interactive study calendar; progress-tracking features (historical performance graphs, achievement badges); and a notification/reminder module delivering push notifications for upcoming assessments, daily study tasks, and newly recommended resources.

(vi) Parent Portal:

An integrated Parent Portal enables parents to view their child's learning plans and mark weekly plan adherence through follow-up reports, which are visible to teachers through the dashboard. This creates a three-way accountability loop between student, teacher, and parent.

4. RESULTS AND DISCUSSION

A. Experimental Setup

The system was deployed and evaluated in two secondary schools in Sri Lanka's Western Province offering O/L ICT programmers. The study involved 120 Grade 11 students (60 per school) and six ICT teachers over one academic term (three months). Participants were divided into a control group (n=60, traditional methods) and a treatment group (n=60, AI-driven LMS). Performance was assessed via third-term examinations, five individual assignments, and two group-based projects. Teachers completed a two-day training program; students received a mobile app orientation session.

B. Component 1: Automated Grading Performance

OCR and grading evaluation results are summarized below:

Metric	Value
Character Error Rate (CER)	4.3%
Word Error Rate (WER)	8.1%
CER with VLM fallback	3.2% (96.8% accuracy)
Mean Absolute Error (MAE) vs. human graders	1.8 marks (out of 10)
Pearson Correlation with human graders	0.89
Agreement within ±2 marks	87.3%
Agreement within ±1 mark	68.5%
Grading time (manual)	15.2 min/sheet
Grading time (automated)	2.3 min/sheet (84.9% reduction)
Teacher's weekly time savings	12.5 hours/week

Table 2. Component 1 — OCR and Automated Grading Evaluation Results

Feedback quality was rated by teachers (n=6) on a 5-point Likert scale: Relevance 4.2/5.0, Constructiveness 3.9/5.0, Language Clarity 4.5/5.0.

C. Component 2: Knowledge Graph and Recommendation Performance

Model	**Precision**	**Recall**	**F1-Score**
Rule-Based Baseline	0.64	0.58	0.61
Collaborative Filtering	0.71	0.68	0.69
BKT Only	0.78	0.75	0.76
GNN + BKT (Proposed)	**0.85**	**0.82**	**0.83**

Table 3. Component 2 — Learning Gap Prediction Model Comparison

Recommendation effectiveness: Teachers evaluated 180 automatically generated recommendations — 68.3% accepted without modification, 22.8% accepted with minor edits, 8.9% rejected. Students following reteaching recommendations demonstrated +18.4% average score increase in weak topics ($p < 0.01$), with 72% progressing from 'Needs Improvement' to 'Proficient' mastery level.

D. Component 3: Question Paper Generation Performance

The Bloom's Taxonomy classification model achieved 86% accuracy on the annotated O/L ICT question dataset, with a fine-tuned DistilBERT achieving 90%+ in the full production configuration. The ILP-based selection algorithm consistently satisfied all teacher-specified constraints across 45 test paper generations, with 100% topic coverage compliance and zero question repetition violations within a three-month window.

E. Component 4: Personalized Learning Plan Performance

Group	**Pre-Test Score**	**Post-Test Score**
Control Group (n=60)	52.3%	58.7% (+6.4%)
Treatment Group — AI LMS (n=60)	51.8%	66.2% (+14.4%)

Table 4. Learning Outcome Comparison — Control vs. Treatment Group

Independent samples t-test: $t(118) = 3.42$, $p < 0.001$, confirming statistically significant improvement favoring the AI-driven LMS treatment group. Mobile app engagement metrics: average 4.2 weekly sessions, 18.3-minute average session duration, 91.7% progress tracking adoption, 86.7% resource access rate, 68.3% study planner adoption.

F. Challenges Identified

- Initial OCR errors with highly inconsistent handwriting (5–10% of cases requiring manual review).
- Limited availability of quality Sinhala/Tamil language learning resources in the recommendation database.
- Some students lacked consistent internet access for mobile app features.
- Teacher learning curve for knowledge graph interpretation (approximately 2–3 weeks for proficiency).

5. CONCLUSION

This research presented TeachMate AI, an integrated AI-Driven Learning Management System addressing critical gaps in Sri Lankan O/L ICT education. By combining VLM-based automated grading, BKT and GNN-powered knowledge graph analytics, Bloom's Taxonomy-aligned intelligent question generation, and personalized adaptive learning plans, the system creates a comprehensive feedback loop supporting both teachers and students.

Experimental evaluation with 120 students and 6 teachers over one academic term demonstrated: 84.9% reduction in grading time while maintaining 87.3% agreement with human evaluators; 0.83 F1-score in predicting learning gaps using the GNN+BKT model; 86% accuracy in Bloom's Taxonomy classification for automated question generation; and a statistically significant 14.4% learning improvement for students using personalized plans versus 6.4% for the control group ($p < 0.001$). These findings confirm that AI-driven automation and personalization can significantly enhance educational effectiveness and efficiency in resource-constrained contexts.

Future work will focus on: extending multilingual support to Sinhala and Tamil; incorporating offline-capable mobile features; integrating additional assessment formats, including diagrams and code; longitudinal evaluation across multiple academic years; and exploring federated learning approaches to preserve student privacy while enabling cross-school model improvement.

REFERENCES

[1] Sharma S, Gupta R. A Smart Assistant for Mental Health and Productivity Enhancement. International Journal of Engineering Research & Technology (IJERT). 2022;11(3):200–205.

[2] Almarashdeh M, Alsmadi MK. How to make them use it? Citizens' acceptance of M-government. Applied Computing and Informatics. 2017;13(2):194–199.

[3] Balfour D. Can computers grade student essays? Educational Leadership. 2013;71(1):70–73.

[4] Page E. The imminent grading of essays by a computer. Phi Delta Kappan. 1966;47:238–243.

[5] Ramesh D, Sanampudi SK. An automated essay scoring system: a systematic literature review. Artificial Intelligence Review. 2022;55:2495–2527.

[6] Sung C, Dhamecha T, Saha S, Ma T, Reddy V, Arora R. Pre-training BERT on domain resources for short answer grading. In: Proc. EMNLP; 2019. p. 6071–6075.

[7] Smith, R. An overview of the Tesseract OCR engine. In: Proc. ICDAR, vol. 2; 2007. p. 629–633.

[8] Kumar M, Jindal RK, Jindal MK, Sharma RK. Handwritten text recognition: a survey. International Journal of Information Technology. 2021;13:2265–2276.

[9] Chen Y, Liu Q, Huang G, et al. Tracking knowledge proficiency of students with educational priors. In: Proc. ACM CIKM; 2017. p. 989–998.

[10] Corbett AT, Anderson JR. Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction. 1995;4(4):253–278.

[11] Nakagawa H, Iwasawa Y, Matsuo Y. Graph-based knowledge tracing. In: Proc. IEEE/WIC/ACM WI; 2019. p. 156–163.

[12] Limongelli C, Sciarrone F, Temperini M, Vaste G. The Lecomps5 framework for personalized web-based learning. Computers in Human Behavior. 2011;27(4):1310–1320.

[13] Mousavinasab E, et al. Intelligent tutoring systems: a systematic review. Interactive Learning Environments. 2021;29(1):142–163.

[14] Thai-Nghe N, Drumond L, Krohn-Grimberghe A, Schmidt-Thieme L. Recommender system for predicting student performance. Procedia Computer Science. 2010;1(2):2811–2819.

[15] Zhang J, Hao X, Sohail MA, Chen L. A survey on personalized learning recommendation systems. In: Proc. AIAM; 2019. p. 428–432.

[16] Crompton H, Burke D. The use of mobile learning in higher education: A systematic review. Computers & Education. 2018;123:53–64.

[17] Anderson LW, Krathwohl DR. A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy. New York: Longman; 2001.

[18] Yahya AA, Toukal Z. Bloom's taxonomy-based classification for item bank questions using SVMs. In: Modern Advances in Intelligent Systems and Tools. Springer; 2012. p. 135–140.

[19] Goel A, Singh G. Automatic classification of questions into Bloom's cognitive levels using transformer-based models. Education and Information Technologies. 2023;28:5649–5670.

[20] Allen IE, Seaman J. Digital Learning Compass: Distance Education Enrollment Report. Babson Survey Research Group; 2017.

[21] Burrows S, Gurevych I, Stein B. The eras and trends of automatic short-answer grading. International Journal of Artificial Intelligence in Education. 2015;25:60–117.

AUTHOR BIOGRAPHIES

Pathiraja P.U.M. is an undergraduate researcher in the Faculty of Computing, SLIIT, Malabe, Sri Lanka. His research interests include computer vision, OCR systems, and AI-powered educational technology. He led the development of the VLM-based grading component in this project. Email: ushanmihiranga2017@gmail.com

Wanniarachchi W.A.P.M. is an undergraduate researcher at SLIIT focusing on knowledge representation, graph neural networks, and adaptive learning systems. He led the development of the knowledge graph and BKT/GNN reteaching guidance module. Email: malshanwanniarachchi001@gmail.com

Hettiarachchi R.H. is an undergraduate researcher at SLIIT specializing in NLP and automated assessment. She led the development of the Bloom's Taxonomy-aligned question paper generation component. Email: ridmihansini97@gmail.com

Jayasooriya L.T. is an undergraduate researcher at SLIIT with interests in mobile learning, personalized systems, and recommendation engines. She led the development of the personalized learning plan generator and mobile application. Email: lashinijayasooriya3@gmail.com