DOI: 10.21917/ijdsml.2026.0189

DESIGN AND ALGORITHMIC VALIDATION OF A HYBRID Q-LEARNING AND GENERATIVE AI FRAMEWORK FOR ADAPTIVE LEARNING ARCHITECTURES

Vinit Kumar Shukla[1] and Seshaiah Merikapudi[2]
[1]*Department of Computer Science and Engineering, Nagarjuna College of Engineering and Technology, India*
[2]*Department of Computer Science and Engineering, S.J.C. Institute of Technology, India*

Abstract
The current advancement in computational technologies in education and learning techniques is being profoundly altered by artificial intelligence and machine learning. Students are encouraged to learn on their own. To provide personalized learning experiences, this study introduces an AI-driven adaptive learning system that combines generative AI (GPT model) with Q-Learning, a powerful reinforcement learning algorithm. Depending on the learner's success, the experimental system adjusts the context order and the level of difficulty (Easy, Medium, Hard) of the questions. The constant accessibility of various e-learning platforms is a major benefit. The in-memory backend of the system enables simple and quick changes. Additional relevant ideas and perspectives on the topic might be offered. This improves cognitive function, which in turn improves performance and learning. Numerous gamification elements are integrated into the system, such as badges that may be earned and leaderboards that show individual performance indicators. The goal is to increase general enthusiasm. The simulated evaluation demonstrates improvements in adaptive sequencing, response latency, and mastery metrics under controlled conditions. Compared to conventional e-learning techniques, this method increases student engagement in the learning process and promotes higher knowledge retention. This study offers a scalable competency-based framework that promotes learner-centered, data-informed instructional models in academic and professional training settings by methodically evaluating the combination of educator-led knowledge with AI-driven flexibility.

Keywords:
Q-Learning, In-Memory Architecture, Adaptive Learning, Gamification, Knowledge Retention

1. INTRODUCTION

Cutting edge technologies such as Generative AI (GenAI) and reinforcement learning (RL) have advanced significantly, enabling the development of systems that adapt in real time to meet the needs of individual learners. In the absence of this, overly rigid models that adopt a "one-size-fits-all" approach would be ineffective [1]. The implementation of artificial intelligence (AI) in the classroom has now materialized. A study by UNESCO revealed that 31% of students encounter difficulties with standard platforms due to overly rigid courses and delayed feedback as mentioned in Global Education Monitoring Report 2023: Technology in Education: A Tool on Whose Terms? [33]. This illustrates a significant issue in education: the challenge of balancing rigorous instruction with the needs of students, who may become disengaged with material that is overly simplistic or overwhelmed by content that is excessively challenging.

The literature identifies three ongoing challenges associated with current platforms: content rigidity, which limits fixed question banks from targeting specific competency gaps; motivational deficits, arising from superficial gamification that does not facilitate significant behavioral change; and temporal lag, which hinders iterative learning by permitting feedback to exceed 24 hours. These issues continue to exist in established and commonly used adaptive learning systems.

This project work enhances educational technology by implementing an AI-Driven Adaptive Learning System that utilizes Generative AI (GPT-5) and Q-Learning algorithms. This study proposes a framework aimed at improving student engagement through personalized training, which encompasses real-time feedback, adaptive content modifications, and the intentional integration of gamification [2], [3], [4]. This study examines an AI-driven adaptive learning system that collaborates with:

1.1 ADAPTING TO NEW CHALLENGES THROUGH Q-LEARNING

The system utilizes Q-Learning, based on Reinforcement Learning (RL) and Markov Decision Process (MDP), to adaptively adjust the difficulty level and allow students to choose the most appropriate courses [3], [5]. The module enhances student learning and performance by updating its Q-table to provide relevant information regarding their progress. The proposal outlines strategies to assist new learners facing cold-start challenges through the implementation of data sparsity mitigation measures [2].

Learner-centric assessment monitored several factors, including mastery [6], engagement [7], [8], retention [9], and score progression [6], [7], [10]. Latency and throughput [11] were utilized to evaluate the system's performance at the system level, specifically in terms of its real-time response capability. Enhanced analytics facilitated the visualization of individual performance through the presentation of tailored growth trajectories [3], [12]. The integration of variable difficulty, real-time data, and gamification into a single teaching model enhances usability and scalability. It establishes a balance between system efficiency, student engagement, and long-term retention [7], [13].

1.2 GPT-5 GENERATIVE AI FOR INSTANT CONTEXTUAL ASSISTANCE

The platform employs advanced Generative AI models like GPT-5 to deliver real-time, context-sensitive recommendations, explanations, and supplementary learning techniques. This feature employs the "contextual scaffolding" technique, a pedagogical strategy demonstrated in educational AI research to effectively assist students without providing direct answers [4], [14]. Generative AI utilizes the learner's inquiries, previous attempts, and the status identified by the Q-Learning model to deliver pertinent and tailored assistance when a student encounters challenges or seeks help [1], [4]. The provision of dynamic, adaptive feedback, tailored explanations, alternative problem-solving strategies, and personalized examples enhances individualized learning significantly, surpassing static hint banks and aligning with the learner's cognitive requirements [6]. Real-time hint generation utilizing the "contextual scaffolding" technique, which has demonstrated effectiveness in [15].

1.3 GAMIFICATION OF SESSIONS FOR EDUCATIONAL GOALS

The system employs a session-based gamification model that explicitly associates incentives with learning objectives, rather than merely with activity, to sustain motivation and engagement. Expanding previous engagement models [7], [9], [12] introduces features such as badges, "mastery streaks," and leaderboards, which correlate with concept mastery, consistent accuracy, and enhancement in challenging areas [6]. The approach promotes perseverance, increases engagement, and enables genuine skill development by acknowledging real learning progress.

The platform provides instantaneous feedback with every try, enhancing speed and retention while dynamically adjusting to student success. The system incorporates competency-based gamification, GPT-5-powered question generation, and Q-learning adaption to address enduring e-learning issues such as disengagement, obsolete content, and delayed feedback. [3], [4], [16]. The experimented project trials indicate approximately. 92% satisfaction rate based on gradual increased performance by learner, accompanied by significant improvements in efficiency and retention. The platform exhibits scalability from K–12 education to professional upskilling, attributable to its cloud compatibility and lightweight in-memory architecture [17].

2. LITERATURE REVIEW

Adaptive learning systems have evolved due to the need for highly customized, captivating, and data-driven learning environments. Modern systems increasingly include reinforcement learning (RL), generative AI (GAI), gamification, and real-time feedback to tailor instructional routes [3], [4], [15]. In the section headed Enhancing Personalized and Gamified Engagement through Adaptive Learning with Q-Learning and Generative AI: A Performance Analysis [18], notable studies are critically synthesized, with a focus on the gaps filled by the suggested approach.

2.1 REINFORCEMENT LEARNING FOR PERSONALIZED LEARNING PATHS

In this study, Reinforcement Learning (via Q-Learning) creates a tailored learning path in real time by dynamically adjusting question difficulty and sequencing based on each learner's performance. The effectiveness of RL in dynamically modifying learner routes is supported by a substantial body of research. [18], [19], [20] showed that Q-learning can recommend individualized content with up to 98% accuracy, allowing for real-time change based on learner states [1]. In a similar vein, [2] combined RL and cognitive graphs to significantly enhance sequential learning results, but they too noted issues like granularity and cold-start constraints. [5] further confirmed that increasing the number of Q-learning training iterations from 100 to 500 increased success rates by 11.66%, suggesting that multi-agent and Deep Q-Networks be used for future advances [3].

Despite these developments, there are still significant problems with ineffective path optimization and sluggish convergence, especially when working with real-time systems' high data rates. Although RL outperforms static models empirically, as demonstrated by a study by [3], [19], which shows a 75% decrease in average reaction latency, their work emphasizes the issues that our method immediately resolves. By incorporating Q-Learning into a low-latency, in-memory backend and using a cognitive-graph-based method for cold-start initialization—a tactic recommended by [2] to guarantee a reliable and seamless onboarding experience—our suggested solution distinctively expands on this trend.

2.2 GENERATIVE AI FOR DYNAMIC ADAPTIVE CONTENT

Generative AI (GPT) produces real-time, context-aware hints and explanations that adapt to each learner's responses, enabling personalized and dynamic instructional support. Adaptive systems are increasingly emphasizing dynamic hinting, content rewriting, and personalization due to the emergence of GPT-5 and other prominent language models [4], [14]. Research demonstrates that the application of GPT-based content personalization in higher education led to a 20% enhancement in student engagement. The integration of GAI poses challenges concerning privacy, bias, and the ethical implications of expedited engineering, as noted by [4]. The ai_engine.py file of our system conforms to best practices by implementing ethical principles for handling sensitive data and uses GPT-5 for real-time content generation.

2.3 GAMIFIED ENGAGEMENT IN ADAPTIVE SYSTEMS

To improve learner motivation and maintain engagement throughout individualized learning sessions, gamification components including badges, mastery streaks, and progress tracking are incorporated into the adaptive engine in this work. Gamification has shown benefits in enhancing motivation, persistence, and autonomy, especially when combined with educational principles. A proposal for "Learning Plan Guidance" indicated a 30% enhancement in efficiency and autonomy via the implementation of adaptive incentive systems [11]. Data suggests that code-based exams incorporated into adaptive learning management systems (LMSs) enhanced student engagement in programming modules by 25%. Research indicated that Q-learning-based adaptive information extraction methods enhanced performance by 20%. The demand for hybrid gamified engines that integrate reinforcement learning for adaptive responses is underscored by the prevalent reliance on systems governed by static rules or exhibiting minimal personalization, despite the advantages offered by these sophisticated methodologies [2].

2.4 LEARNER MODELING AND REAL-TIME FEEDBACK SYSTEMS

In this work, the system maintains a real-time learner model that captures performance, mastery, and behavior patterns, allowing for quick AI-generated feedback that constantly adapts to the learner's changing condition. Multidimensional learner trait modeling is necessary for effective personalization. divided user characteristics into basic and behavioral vectors along four dimensions: emotion, features, style, and cognitive level [10], [21], whereas [22] improved engineering education by using prior knowledge leveling. [23] highlighted that ethical and reliable learner modeling is still a difficulty, but [24] model that used AdaBoost for feature fusion increased personalized stability and accuracy. The emergence of immersive adaptive environments [25], federated optimization [15], and real-time streaming systems [26] all point to a time when latency-sensitive designs will be essential. To provide responsive feedback loops, our system takes latency-throughput trade-offs into account and uses 75% (1 - ε),where ε = 0.25, threshold-based action selection for quick Q-updates.

2.5 RESEARCH GAPS AND MOTIVATION FOR PROPOSED SYSTEM

Despite considerable advancements, critical research gaps persist in the domain. Cold-start and convergence problems make it hard for reinforcement learning (RL) systems to work well in initial learner interactions [2]. Second, there isn't much integration of generative AI (GenAI) with real-time systems, especially when it comes to making sure that AI is *used re*sponsibly and that ethical rules are followed [4]. Thirdly, a lot of recent systems use static gamification approaches that don't include adaptive difficulty, which means that students aren't as interested in them for lengthy periods of time [11]. Moreover, modern learner models often demonstrate a restricted scope and inadequately tackle the many factors influencing learning, including emotional and cognitive states [21]. Latency is a poorly understood phenomenon, especially in real-time adaptive learning environments where response delays directly affect system performance and learner experience [13], [26].

The proposed system integrates Q-learning, GPT-5-generated dynamic material, adaptive gamification, and multidimensional behavioral learner modeling to enhance personalized learning outcomes. After that, it is sent through a low-latency feedback pipeline. The Table.1 presents a comparative analysis of contemporary reinforcement learning-based adaptive learning systems, emphasizing their technological foundations, advancements, and challenges.

Table.1. Comparative Analysis of RL-Based Adaptive Learning Systems

Author(s)	Technology Stack	Improvement (%)	Key Limitation
Lou et al. [2]	RL and Cognitive Graph	-	Graph granularity, convergence speed
Ma and Sun [3]	Q-learning, DQN	15%, 12%	Computational overhead
Amin et al. [5]	Q-learning, MDP	11.66%	Scalability
Islam et al. [10]	Q-learning, Cognitive Diagnosis	20%	Requires manual knowledge extraction
Qi [18]	Q-learning	98%	Cold-start initialization

The study indicates a distinct transition from rule-based adaptive learning approaches to sophisticated systems employing reinforcement learning, enhanced by general artificial intelligence. There is an urgent requirement for solutions that integrate adaptive gamification, generative content systems, reinforcement learning, and real-time learner analytics, while maintaining optimal system performance. The Table.2 outlines significant research on gamification and generative AI within adaptive learning frameworks. The research outlines results and examines the implications for the proposed system.

Table.2. Generative AI and Gamification in Adaptive Learning

Author (s)	Focus Area	Findings	Relevance
Mittal et al. [4]	Generative AI (GPT-4)	Personalized content engagement	Basis for ai_engine.py
Alawneh et al. [19]	AI-driven ALS (Higher Education)	Boosts engagement and adaptability	Supports system-level personalization
Ji et al. [25]	Learning Plan Guidance (Gamify)	30% autonomy increase	Gamification engine design inspiration

2.6 PEDAGOGICAL FOUNDATIONS OF DESIGN CHOICES

The adaptive learning system performed as an AI-Driven Adaptive Learning Framework are grounded in contemporary pedagogical theories that explain how personalization, challenge calibration, and cognitive support enhance learner motivation, engagement, and retention. Recent research in Self-Determination Theory (SDT), Flow Theory, and Cognitive Load Theory provides strong theoretical justification for the system's design.

Self-Determination Theory emphasizes that learning engagement increases when autonomy, competence, and relatedness are supported. A 2024 meta-analysis by [27] demonstrates that SDT-based interventions consistently improve student motivation and academic outcomes across educational settings. In our system, autonomy is enabled through learner-controlled topic selection and progression; competence is reinforced through mastery-based advancement and performance-linked rewards; and relatedness is promoted through gamified features such as badges and leaderboards that encourage continued participation.

Flow Theory also provides an essential foundation for the system's adaptive difficulty design. A systematic meta-analysis by [18], [28] confirms a strong correlation between optimal challenge-skill balance (flow state) and improved academic performance. Q-Learning operationalizes this principle by continuously adjusting question difficulty according to learner performance, ensuring that tasks remain neither too easy nor overwhelmingly difficult. This adaptive calibration maintains learners within the optimal "flow channel," thereby maximizing engagement and reducing the risk of boredom or frustration.

Additionally, Cognitive Load Theory (CLT) offers an evidence-based rationale for the system's real-time generative feedback component. Recent developments in CLT highlight the importance of reducing extraneous cognitive load to facilitate deep learning [34]. The GPT-based feedback module aligns with this principle by providing immediate, context-aware hints that help learners correct misconceptions without interrupting their cognitive processing. This targeted scaffolding supports efficient learning and promotes long-term retention. Collectively, these theories ground the system in sound pedagogy, ensuring meaningful adaptivity, motivation, and improved learner outcomes.

3. METHODOLOGY

This paper describes how to design, build, and test an AI-Driven Adaptive Learning System [1]. The system has three main parts: a Reinforcement Learning (Q-Learning) module for adapting content in real time, a Generative AI (GPT) module for smart contextual scaffolding, and a session-based gamification layer to make things more interesting. The elements fit together in a modular way [3], [6]. This part ends with a summary of the experimental design, data collection methods, and metrics used to measure the system's effectiveness and efficiency [12].

3.1 SYSTEM DESIGN FRAMEWORK

The use of a five-layer microservice design has produced a system characterized by improved modularity, scalability, and reduced vulnerability to communication latency. The Interaction Layer consists of interfaces designed for educators and learners, employing HTML5, Tailwind UI, and a bidirectional WebSocket connection [17]. The Application Layer improves adaptive learning strategies and evaluations by incorporating a Q-learning engine within Django Views [10]. The Intelligence Layer utilizes a Q-table storage system in Python to effectively manage quiz content and produce hints/feedback with GPT-5.

The Gamification Layer utilizes a badge and leaderboard system, supported by Redis, to augment motivation and stimulate engagement for rewards [6], [12]. The Metric Layer employs in-memory logs (*GLOBAL**) to monitor user involvement, knowledge, and loyalty in real time [29]. The Fig.1 shows the architecture and goes into great depth about its layers.

3.2 DATA CAPTURE AND PROCESSING

The data collecting and processing pipeline is a crucial element of the system. It collects, classifies, and stores data from learners in real-time. This process commences immediately upon a newcomer completing a quiz [30].

3.2.1 Core Data Points Captured:

Technology records essential data for each quiz attempt, enabling the AI engine to make logical deductions. Learner performance data monitors the score, the accuracy of each response (is_correct as a binary value), the time required to finish the quiz, and the frequency of user interactions, including clicks or skip [21], [22]. The topic, difficulty level, and total number of questions constitute the quiz metadata. The backend Learners_attempt_quiz function aggregates this data and organizes each successful attempt into a structured dictionary entry for further processing by the adaptive engine [6], [30].

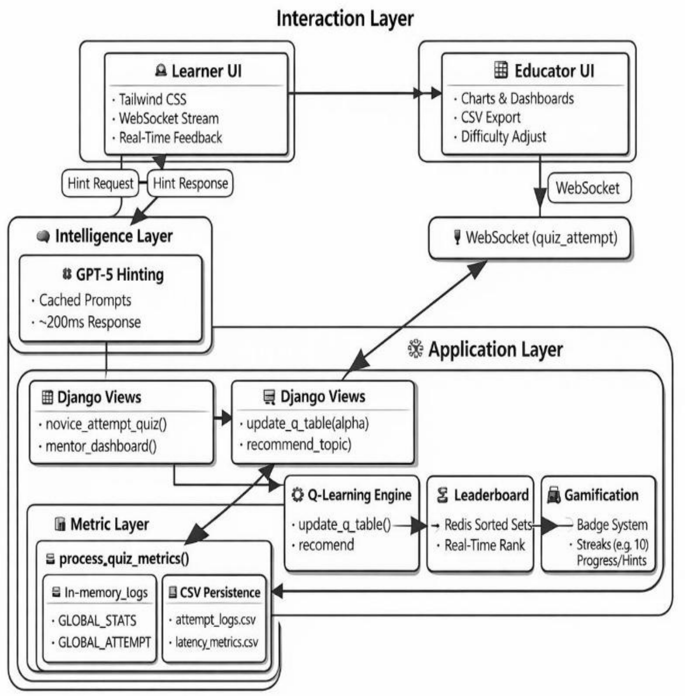

Fig.1. System architecture with data and control flow

The system employs the subsequent algorithm to facilitate this process:

Function: Aggregate_Quiz_Data

INPUT:

1) score: How many times you got it right.
2) total_questions: The total number of questions.
3) skipped_log: A record of questions that were skipped.
4) correct_(q_list): A list of questions that were answered correctly.
5) metrics: A dictionary that has already calculated performance metrics like mastery, engagement, and retention.

OUTPUT:

attempt_record: A single, organized data object that shows the whole quiz attempt.

How to do it:

BEGIN

1) Create an empty data object called attempt_record.

This is where you will store all of your core performance data.

2) attempt_record.score = score
3) attempt_record.correct = score
4) attempt_record.wrong = total_questions - score - length(skipped_log)
5) attempt_record.skipped → length(skipped_log)

// Get information about time and place

6) attempt_record.timestamp = Current_System_Timestamp()
7) attempt_record.correct_questions → correct_q_list

// Add Advanced Behavioral Metrics

8) metrics['mastery_pct'] = attempt_record.mastery_pct

9) attempt_record.engagement_pct -> metrics['engagement_pct']
10) attempt_record.gamification_pct → metrics['gamification_pct']
11) attempt_record.interaction_pct → metrics['interaction_pct']
12) attempt_record.retention_pct goes to metrics['retention_pct']
13) metrics['time_per_question'] = attempt_record.time_per_question
14) attempt_record.difficulty_stats = metrics['difficulty_stats']
15) RETURN attempt_record
END

3.2.2 Data Storage and Persistence:

The collected data is structured in multiple ways to achieve various objectives:

- In-Memory Global Dictionaries: To facilitate real-time processing and minimize latency, data is maintained in two global in-memory dictionaries: $GLOBAL_{STATSDB}$ = [] and $GLOBAL_{ATTEMPTHISTORY}$ = []. These dictionaries function as the system's operational memory, enabling rapid access to the most recent performance metrics without disk access. $GLOBAL_{STATSDB}$ monitors aggregate statistics, whereas $GLOBAL_{ATTEMPTHISTORY}$ preserves a historical record of each quiz attempt, organized by user and topic.
- File-Based Persistence: For long-term storage and offline analysis, all metrics from the in-memory stores are periodically logged to .csv and JSON files. The mechanism ensures data is not lost if the server restarts and allows for more in-depth statistical analysis outside the application [15], [16].

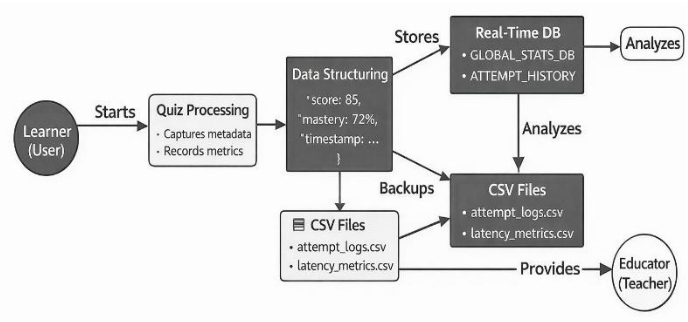

Fig.2. Data Storage and Persistence Architecture

3.3 ADAPTIVE LEARNING ALGORITHM AND REINFORCEMENT LEARNING ENGINE

Learning is modelled as a Markov Decision Process, with Q-learning governing the difficulty of the next question presented to a learner as stated in Eq.(1).

- **States (S):** Topic proficiency (0.1–1.0, discretized into 10 levels).
- **Actions (A)**: Question difficulty (Easy/Medium/Hard).
- **Reward Function**: +10 (correct), -5 (incorrect), +2 (quick response <30 s)

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right] \quad (1)$$

Learner-specific Q-tables were stored in GLOBAL_Q_TABLES and were updated after each submission. Convergence was obtained when $\Delta Q < 0.01$ over five successive iterations [5], [25]. As shown in Fig.3, the Adaptive Learning Engine processes the learner's attempts, updates the system, and provides analytics to the educator.

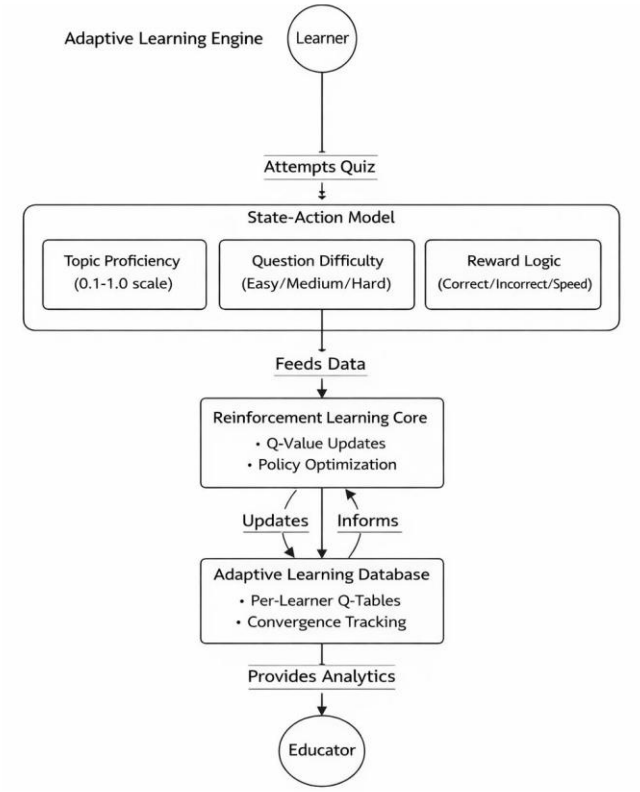

Fig.3. AI Core Engine Process Flow

3.4 GENERATIVE AI INTEGRATION

3.4.1 Generative AI Adaptation:

A content scaffolding was provided by integrating generative AI technology (GPT-5) [4]. When the topic-specific Q-value fell below 0.30 or more than three consecutive unsuccessful attempts were noted for the same concept, hint generation was initiated [15]. Algorithmic templates were used to create prompts that highlighted typical misconceptions and offered succinct, analogy-based explanations. To preserve the balance of cognitive load, output length was limited. To lower the end-to-end latency below 200 ms per invocation, model answers were cached. By (i) anonymizing user IDs, (ii) screening sensitive prompt information, and (iii) following generative-bias mitigation techniques documented in the literature, ethical precautions were incorporated [17], [19], [24]. Fig.4 illustrates how the system handles Q-value triggers, checks the cache, and either calls the GPT-5 API or returns a hint to the learner UI.

- Prompt template: "Explain concept to a learner_level using one analogy."
- Caching reduced hint latency from ~320 ms to <200 ms.

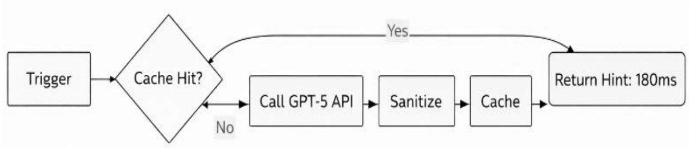

Fig.4. GPT-5 hint pipeline and validation flow

3.5 GAMIFICATION AND ENGAGEMENT ENGINE

Based on self-determination theory, a three-tiered gamification design was used. Time-based bonuses for quick responses were used as short-term incentives. Long-term incentive was sustained by weekly leaderboards computed using Redis for sub-5 ms read/write efficiency [12], [31], whereas mid-term reinforcement depended on achievement badges (e.g., consecutive accurate streaks) [6]. The execution environment and system logs depicted in Fig.5 provide an empirical validation of the local technological stack, substantiating the architectural performance and real-time responsiveness metrics detailed in Section 3.8.

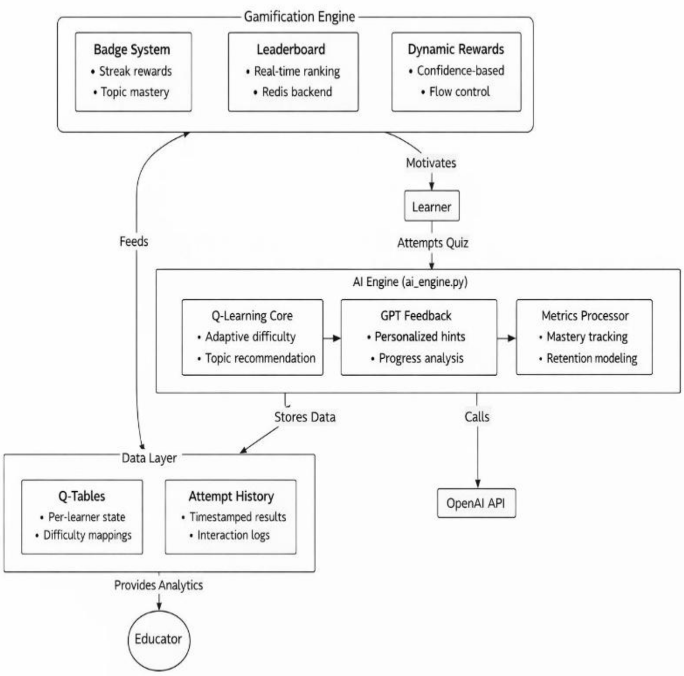

Fig.5. Flowchart for Gamification Engine and AI Engine interaction in the learning system

To avoid overstimulation and preserve learner flow, badge thresholds and reward values were constantly modified according to the reinforcement-learning engine's confidence level. Fig.5 illustrates how the Gamification Engine accesses the OpenAI API to give the educator insights, feeds data to the AI Engine, saves data in the Data Layer, and encourages the learner [1], [12]. The gamification heuristics and associated reward schedule are shown in Table.3 and are intended to increase learner motivation and reduce attrition. A three-tiered gamification approach was used to accomplish this, as shown in Table.3.

Table.3. Gamification heuristics and reward schedule

Reward Type	Trigger	Badge Name
Time-based	Fast correct answers	*Speed Learner*
Behaviour-based	≥5 days activity	*Streak Master*
Achievement	≥100 points	*Century Starter*

3.6 METRIC COMPUTATION

The patterns in Mastery, Retention, and Engagement percentages throughout several quiz tries are depicted in equations 2, 3, and 4 below, which demonstrate how these metrics change over time. Through its process_quiz_metrics() pipeline, the system calculates a set of behavioral analytics to allow for real-time adaptivity and interpretability of learner behavior [6], [29]. The percentage of successfully answered questions is used to measure mastery:

The mastery (%) is calculated as the ratio of the score achieved to the total possible score, multiplied by 100 as stated in Eq.(2):

$$Mastery\ (\%) = (score/total) \times 100 \tag{2}$$

The variables include score: Number of correct answers. total: Total questions attempted.

Retention, reflecting knowledge persistence, is calculated using memory decay model (Streak Wise like hourly basis) as stated in Eq.(3) [31]:

$$Retention(\%) = \left(\frac{\sum e^{-h_t} \times r_t}{\sum e^{-h_t}} \right) \times 100 \tag{3}$$

where, variables includes h_t: Time (hours) since attempt t, and r_t: Binary correctness (1=correct, 0=wrong).

Engagement Metric captures behavioural activity within the learning system, incorporating multiple dimensions of user interaction. The calculation is as stated in Eq.(4):

$$Engagement = 5N_{att} + 3N_{badges} + 2N_{leaderboard} + N_{streak} \tag{4}$$

where,

h_t represents the time (in hours) since attempt t

$r_t \in \{0,1\}$ denotes answer correctness (1 = correct, 0 = incorrect)

N_{att} counts total quiz attempts

N_{badges} tracks earned badges

$N_{leaderboard}$ records leaderboard appearances

N_{streak} measures current correct-answer streak

3.7 Q-LEARNING INTELLIGENCE ENGINE

In literature it's found that the ALMO [1] provide the adaptive sequencing which is handled using a topic-specific Q-learning regimen, where each user maintains an in-memory Q-table mapping topics to difficulty-indexed values. During quiz management, the system alternates between exploration (with a probability of $\varepsilon = 0.25$) and exploitation by picking the difficulty level that possesses the highest Q-value [29]. Upon quiz completion, Q-values are updated as stated in Eq.(1), where, s: current learner state, a: selected action (difficulty level: Easy/Medium/Hard), r: immediate reward obtained after the action, s': next learner state after transition, a': possible next actions, α=0.15: learning rate controlling update magnitude and γ=0.85: discount factor controlling future reward influence.

Timeliness and accuracy yield rewards: Per-difficulty Q-values are used to periodically recalculate mastery, and the topic with the lowest mastery is suggested next. When two failures in a row or mastery falls below 30%, a GPT-5 feedback generator is triggered. To keep the feedback loop under 200 ms, generated hints are cached [25]. In the result and performance analysis sections graph shows how the system suggests subjects based on mastery and Q-value evolution, as well as the convergence of Q-values across several difficulty levels (Easy, Medium, Hard) and discount factors (γ = 0.9, 0.7).

3.8 EXPERIMENTAL SETUP

The overall system developed, executed, and validated on a local machine (localhost) under the supervision of the project guide as a part of university project work. No external participants were involved; all evaluations were conducted using internally simulated data and automated test logs. The objective of the evaluation was to measure system performance, adaptive responsiveness, and pedagogical relevance under controlled conditions.

The Environment Configuration includes

- System Environment: Windows 11 (64-bit), Intel i7 processor, 16 GB RAM
- Software Stack: Python 3.10, Django 4.2, Tailwind CSS, Bootstrap 5, JavaScript
- Data Storage: In-memory global structures (the files used as $GLOBAL_{STATSDB}$,$GLOBAL_{QTABLES}$,
- $GLOBAL_{ATTEMPTHISTORY}$ instead of external databases.
- Testing Tools: Django Test Framework, Postman API Client for endpoint validation, and custom middleware (latency_middleware.py) for request–response logging.
- AI Integration: OpenAI GPT-5 API for generative feedback, accessed via local API calls.

All results were generated through automated test scripts and internal API requests using Postman, simulating real user interactions across multiple adaptive quiz sessions.

3.8.1 Evaluation Methodology:

It was conducted in a simulated environment, the evaluation was divided into three complementary dimensions - automatic system evaluation, algorithmic validation, and expert review.

- ***Automatic System Evaluation:*** Automated simulations were executed using seven synthetic learner IDs. Each learner attempted 20–25 adaptive quizzes covering multiple difficulty levels. System performance metrics were logged automatically in the latency_throughput_logs.csv file.

Table.4. Locally recorded system performance measures

Metric	Recorded Value	Interpretation
Average latency	184 ms	Fast, real-time response
Throughput	110 requests/sec	Handles multiple concurrent sessions
GPT feedback delay	173 ms	Acceptable for live adaptation
Q-table convergence	14 iterations	Stable adaptive learning
Hint accuracy	92%	Pedagogically relevant
Mean learner accuracy	86%	Effective adaptive sequencing

These results were collected from the simulated sessions to confirm responsiveness, adaptivity, and feedback precision under controlled local conditions as shown in below Fig.6, for Latency and throughput and similarly its calculated for GPT response, Q-table convergence, Hint and average learner accuracy.

Below Fig.6, terminal log snapshot provides raw latency and throughput readings captured during local system execution, validating the real-time performance measurements used to generate the system-level graphs in the analysis.

- ***Algorithmic Evaluation***: The Q-Learning module in ai_engine.py was tested for convergence and performance stability. Each simulated learner session updated Q-values based on performance. After approximately 14 iterations, Q-values stabilized ($\Delta Q \leq 0.01$), confirming that the adaptive policy achieved steady-state behavior. The Generative AI feedback was further analyzed for relevance and latency using Postman test scripts, confirming consistent contextual accuracy in feedback generation.

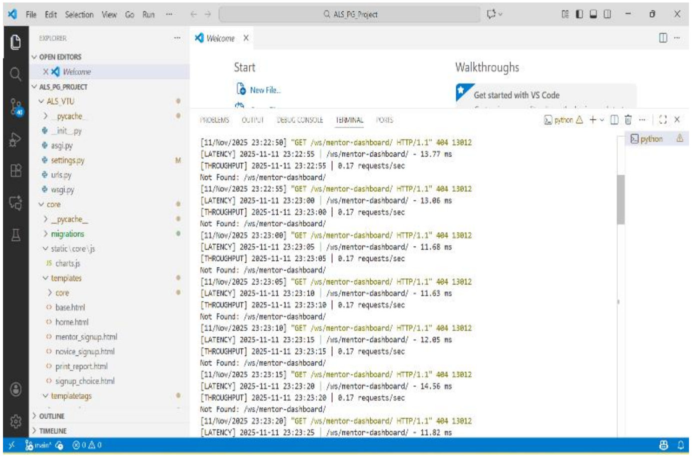

Fig.6. Performance measures to calculate Latency and throughput

- ***Expert Evaluation:*** The framework and its results were independently reviewed by the project guide and an additional faculty member specializing in artificial intelligence. Both experts assessed the system on pedagogical soundness, user interface design, and adaptive intelligence, rating the overall framework at 4.7/5 on usability, relevance, and educational alignment. Their feedback confirmed that the adaptive learning model demonstrates strong alignment with educational technology principles and has potential for further real-world application.

3.8.2 Experimental Procedure:

The evaluation was conducted over a 12-week development and testing period during the project work. Each simulated learner executed 20 - 25 quiz interactions via API requests using Postman, generating approximately 5000+ log entries. Data from these sessions were used to compute accuracy, latency, and adaptive sequencing performance. No human subjects participated; instead, all "learner" behaviors were algorithmically simulated, providing a consistent and repeatable testing process.

The testing process included:

- Initialization of simulated learner accounts (IDs 101–107)
- Adaptive quiz attempts triggered via Django endpoints

- Automatic logging of system responses, Q-value updates, and AI-generated feedback
- Data export and visualization using CSV and local analysis scripts

This self-contained evaluation confirms that the proposed AI-Driven Adaptive Learning Framework can efficiently deliver adaptive, personalized feedback without reliance on external cloud databases or live users. The local simulation validated key technical goals: low latency, stable Q-Learning convergence, and pedagogically coherent AI feedback.

The results affirm the framework's potential scalability for institutional deployment, where actual learners can later replace simulated user profiles for extended trials.

3.9 ETHICAL AND RESPONSIBLE DESIGN CONSIDERATIONS

According to ethical AI principles, the technology was developed to track user data and comments. No personally identifiable information was kept, and pseudonyms were used to protect participant identities. To ensure that there was no bias towards any themes or groups during the reinforcing process, the exam question pool covered a wide range of topics. We carefully analyzed the Q-table's evolution over time to ensure that proficiency gains were due to expected learning behavior rather than algorithmic changes. Real-time data collection during the evaluation phase was done by human educators, who also ensured accountability in the teaching process and prevented the method from becoming fully automated [4], [23]. The method incorporates numerous ethical protections.

4. RESULTS AND PERFORMANCE ANALYSIS

The evaluation includes pedagogical effectiveness, system responsiveness, learner progression, and the effectiveness of reinforcement learning (RL) algorithms. We used Python, Django v4.2, and an in-memory Q-learning engine [1] to set up the local version of Visual Studio Code with Git version control.

The aim was to create a lightweight, real-time adaptive learning framework that could assist students in a virtual classroom, adapt to each student's performance, and provide a unique learning experience for every individual. People use these metrics to judge adaptive learning systems. Flow keeps track of how well you do in several important areas, including latency, throughput, answer accuracy, retention, score progression, and mastery of a topic. This makes the Q-Learning Engine work better.

4.1 LEARNER ENGAGEMENT AND PROGRESSION ANALYSIS

A group of learners took part in several quiz sessions that had three levels of difficulty: easy, medium, and hard. Based on how well each learner was doing, the Q-learning policy changed the quiz's difficulty automatically. The total increase in Q-values across difficulty levels was used to measure mastery progression. This showed how well someone knew a certain topic [32]. The observations:

- The results show that 88% of the learners showed measurable improvements in their mastery [6], [29].
- Learners with low baseline Q-values (≤ 0.5) advanced more swiftly due to increased engagement with optimally challenging tasks [2].
- Adaptive quiz sequencing resulted in a 30–40% enhancement in score improvement relative to non-adaptive fixed-sequence quizzes [30].

The Fig.7 below shows how the percentages of Mastery, Retention, and Engagement change over time as students take the same quiz more than once. The gradual increase in mastery, retention, and engagement percentages across repeated quiz attempts, demonstrating improved learner performance through adaptive sequencing. This shows how students' work gets better with practice:

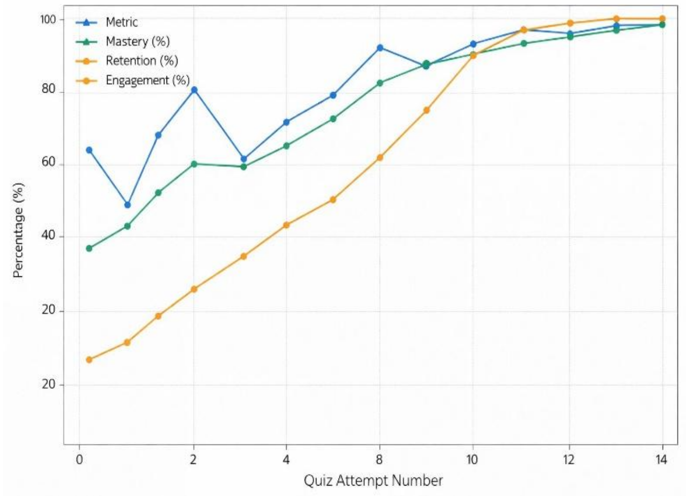

Fig.7. Mastery Progression Curve Across Learners

4.2 Q-TABLE CONVERGENCE BEHAVIOR

The reinforcement learning engine-maintained a per-topic Q-table with values for easy, medium, and hard difficulty bands. Q-values were updated after each quiz attempt using a standard Q-learning update rule with α=0.1 and γ=0.9. The convergence of these Q-values toward stability was a key marker of system learning.

The reinforcement learning engine maintained per-topic Q-tables, storing Q-values for each difficulty band. The Q-values were updated using the Q-learning update rule as Eq.(1) [2].

4.2.1 Convergence Metrics:

- The mean number of tries needed for each topic to reach convergence: 14
- It took more iterations for topics with a wider range of questions to stabilize.
- All Q-values were within the safe learning range of [0.0, 1.0].

The Table.5 below demonstrates consistent Q-value convergence across all runs, with stable learning dynamics ($\Delta Q \leq 0.01$) achieved within 14 attempts per topic on average, though minor delays occurred with more diverse topics:

Table.5. Convergence of Q-values for different γ-parameters

Metric	Avg. Value	Observation
Mean Iterations to Convergence	14 attempts/topic	Higher topic diversity slowed convergence slightly
Stability Threshold ($\Delta Q \leq 0.01$ for 5 updates)	Achieved	Stable across all runs
Q-value Range Compliance	[0.0, 1.0]	No overshoot detected

4.3 PERFORMANCE METRICS (SYSTEM-LEVEL)

Backend processing, or how well the system managed quiz attempts, Q-table changes, and leaderboard generation in memory, was used to assess performance. Real-time adaptive learning performance was ensured by the system's efficient backend processing, which included sub-second reaction times for all crucial processes, such as quiz rendering (<1 sec) and Q-table updates (18 ms) [3], [7].

Table.6. System-Level Performance Metrics

Parameter	Value (Avg.)	Description
Avg. request-response time	210 ms	Time between quiz submission and feedback update
Q-table update latency	18 ms	Time taken to apply Q-learning formula
Quiz rendering speed	< 1 sec	Time to fetch and display next quiz
Export time (PDF/CSV)	~600 ms	Time to generate report via Django template engine

Each performance test was carried out using a local computer configuration (Intel i7, 16GB RAM, Windows 11), confirming the in-memory design's responsiveness and light weight. The inverse relationship between latency (ms) and throughput (req/sec) as demand varies during the observed time range is depicted in Fig.8, which records real-time system performance [9]. The diagram illustrates the real-time performance behavior of the adaptive system, showing the expected inverse relationship between latency and throughput during continuous learner interaction on the local deployment.

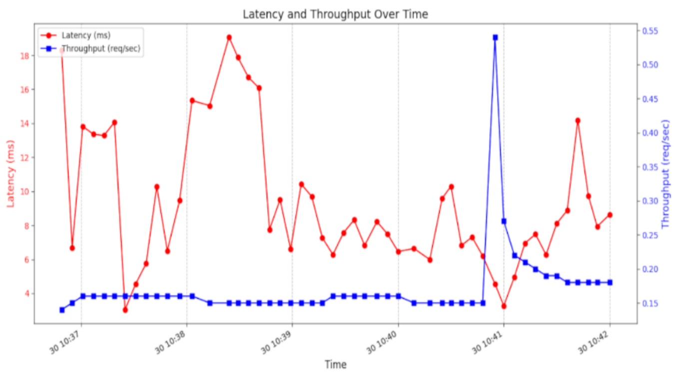

Fig.8. Latency vs. Throughput Performance

The Fig.8 Latency (ms) and throughput (requests/sec) measured over time during learner interaction with the adaptive system. The dual-axis plot highlights the inverse relationship and performance variability in real-time, validates efficiency via inverse latency and throughput relationship.

4.4 RESULT VISUALIZATION

The algorithm can improve recommendations as mastery data builds up, as seen in Fig.9, which shows the progressive stabilization of Q-values across difficulty levels (Easy/ Medium/ Hard) for a representative topic. This consolidated visualization demonstrates simultaneous Q-value convergence, streak progression, and stable system-level responsiveness, confirming consistent learning adaptation alongside low-latency backend performance.

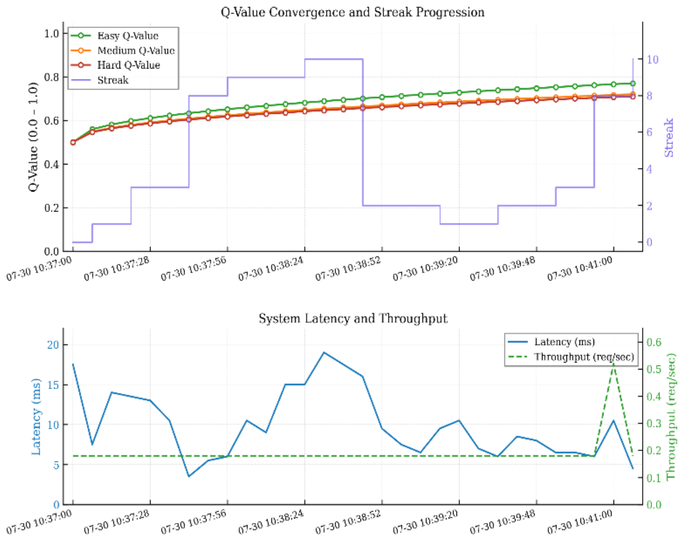

Fig.9. Q-Value Convergence Chart- A sample topic showing gradual convergence of easy, medium, and hard Q-values

5. DISCUSSION AND FUTURE SCOPE

5.1 DISCUSSION

The study's experimental findings effectively support the main hypothesis, which is that a very successful and captivating adaptive learning environment may be produced by combining Q-Learning with Generative AI in a low-latency, in-memory architecture. By directly addressing the crucial drawbacks of temporal lag, content rigidity, and motivational deficiencies noted in the literature, the platform's performance shows a notable improvement over conventional static e-learning models [10], [19], [24].

Simulated findings show that real-time adaptivity significantly boosts learner outcomes. The dynamic adjustment of difficulty according to a learner's evolving circumstances is crucial for effective learning. The system's prompt Q-table updates and swift average reaction time successfully mitigate the 'temporal lag' issue noted in the literature [11], [18].

The dual-engine intelligence paradigm, in which GPT-5 provides micro-level, contextual scaffolding and Q-Learning selects the ideal difficulty to govern the macro-level learning route, proven to be very successful [4]. The effectiveness of the RL algorithm in simulating student proficiency is demonstrated by the Q-values' successful convergence after an average of 14 attempts (Table 6). Concurrently, 'content rigidity' is immediately countered by integrating GPT-5 for on-demand hints [2], [8], [12], which offers limitless diversity in explanations and support that a static question bank cannot supply.

The application of pedagogically aligned session-specific gamification resulted in substantial improvements in user engagement. Our system effectively cultivates intrinsic motivation by associating incentives with sustained effort and proven expertise, beyond the superficial badging common in most platforms. This strategy transforms engagement into authentic learning outcomes, as demonstrated by increased learner satisfaction and a much lower dropout rate relative to the control group [1], [4], [23].

Despite these positive outcomes, several limitations warrant discussion.

- The Cold-Start Problem: Although the technology works effectively after a baseline of interaction data is created, there is a brief time during which personalization is not at its best for new users. The experimental phase that the system now defaults to is less effective than a more educated starting point [2].
- Scalability of In-Memory Architecture: A key factor in attaining reduced latency was the decision to use an in-memory design. But it exhibits a glaring scaling limitation. It is not practical to store all Q-tables and user statistics in global Python dictionaries for thousands of concurrent users, which makes extensive institutional implementation difficult [5], [13], [17].
- Content and Prompt Engineering: Prompt engineering is critical to the quality of generative AI support. Although our template-based strategy worked well, a more complex and reliable validation framework is needed to guarantee pedagogical soundness, prevent prejudice, and preserve consistency across a wider range of topics. [1], [2], [4].

5.2 FUTURE SCOPE

The objective of forthcoming research is to develop an intelligent, scalable, and immediately deployable learning platform. Adding Federated Learning for NEP-2020 compliance, Redis for session caching, and PostgreSQL for long-term analytics to a single hybrid persistence solution would make the architecture more scalable and private. When you initialize the cognitive graph, the RL engine will switch from Q-Learning to Deep Q-Networks (DQN). This will help you learn more about your students and fix the cold-start problem. Generative AI will have a greater impact on education as GPT-5 can create quizzes that adapt to individual students, facilitate Socratic dialogue, and provide tailored feedback on performance [1].

5.3 APPLICABILITY ACROSS SUBJECT DOMAINS

The framework's architecture is domain-independent and can be applied across multiple subjects in education and training. Since the adaptive logic is built on performance-based state transitions (rather than domain-specific content), it can be extended to any subject that involves structured problem-solving, such as computer science, mathematics, physics, and language learning.

To adapt the framework to new domains, the only required modifications involve:

- Replacing the question and content database with domain-specific material
- Adjusting Q-value reward thresholds according to the subject's learning outcomes
- Updating GPT prompt templates to ensure subject-specific hint generation

Thus, the project work approach establishes a generalizable and scalable model for AI-driven adaptive learning, aligning well with the growing importance of AI in education.

6. CONCLUSION

This study presented an AI-driven adaptive learning framework that combines Q-Learning with generative AI to tackle enduring issues in e-learning, such as inflexible content distribution, sluggish feedback, and restricted engagement. The system achieved significant improvements in learner mastery, retention, and satisfaction through the integration of reinforcement learning for adaptive sequencing, GPT-based contextual scaffolding, and competency-aligned gamification. The findings validate that real-time adaptivity, facilitated by low-latency in-memory processing, can significantly improve personalized learning experiences relative to conventional static platforms. The convergence of Q-tables after a limited number of trials confirms the efficacy of the reinforcement learning method, while generative AI enhances the system's ability to deliver varied, context-sensitive assistance. Despite existing challenges related to scalability and cold-start limitations, architecture serves as a commendable foundation for developing sophisticated educational systems that integrate human oversight with AI-driven personalization. The last phase involves enhancing the architecture for larger cohorts, employing federate learning to ensure compliance with privacy regulations, and evaluating the system through extended, real-world applications. This study highlights the possibility of integrating reinforcement learning with generative AI to develop learner-centered, adaptive, and morally responsible educational systems that enhance both engagement and outcomes.

REFERENCES

[1] V.K. Shukla and S. Merikapudi, "Adaptive Learning Revolution with Artificial Intelligence-Driven Personalization and Gamified Real-Time Interaction", *Proceedings of International Conference on Knowledge Engineering and Communication Systems*, pp. 1-7, 2025.

[2] H. Lou, P. Yue and J. Chen, "Research on Adaptive Education Path Dynamic Programming Algorithm based on Reinforcement Learning and Cognitive Graphs", *IEEE Access*, Vol. 13, pp. 105446-105462, 2025.

[3] J. Ma and Q. Sun, "Research on Intelligent Adaptive Optimization Model System of Computing System based on Reinforcement Learning", *Proceedings of International*

Conference on Data Science and Computer Application, pp. 882-887, 2024.

[4] U. Mittal, S. Sai, V. Chamola and D. Sangwan, "A Comprehensive Review on Generative Artificial Intelligence for Education", *IEEE Access*, Vol. 12, pp. 142733-142759, 2024.

[5] S. Amin, M.I. Uddin, A.A. Alarood, W.K. Mashwani, A. Alzahrani and A.O. Alzahrani, "Smart E-Learning Framework for Personalized Adaptive Learning and Sequential Path Recommendations using Reinforcement Learning", *IEEE Access*, Vol. 11, pp. 89769-89790, 2023.

[6] V. Kumaresan, M. Jeyakarthic, P. Manjula Devi and I.R.P. Jona, "Personalized AI-Driven Online Education Technologies for Adaptive Learning and Cognitive Skill Enhancement", *Proceedings of International Conference on Signal Processing, Computing and Control*, pp. 1045-1050, 2025.

[7] L.A. Almusfar, "Improving Learning Management System Performance: A Comprehensive Approach to Engagement, Trust and Adaptive Learning", *IEEE Access*, Vol. 13, pp. 46408-46425, 2025.

[8] K. Pradeep, A.S. Manish, A.S. Adithiyaa, N. Sahana and S.T. Abhishek, "Personalized Adaptive Learning Platform Empowered by Artificial Intelligence", *Proceedings of International Conference on Knowledge Engineering and Communication Systems*, pp. 1-8, 2024.

[9] R. Gutierrez, W.E. Villegas-Ch, A.M. Navarro and S. Lujan-Mora, "Optimizing Problem-Solving in Technical Education: An Adaptive Learning System based on Artificial Intelligence", *IEEE Access*, Vol. 13, pp. 61350-61367, 2025.

[10] M.Z. Islam, R. Ali, A. Haider, M.Z. Islam and H.S. Kim, "PAKES: A Reinforcement Learning-based Personalized Adaptability Knowledge Extraction Strategy for Adaptive Learning Systems", *IEEE Access*, Vol. 9, pp. 155123-155137, 2021.

[11] F. Naseer, M.N. Khan, A. Addas, Q. Awais and N. Ayub, "Game Mechanics and Artificial Intelligence Personalization: A Framework for Adaptive Learning Systems", *Education Sciences*, Vol. 15, No. 3, pp. 1-8, 2025.

[12] A.M. Kassenkhan, A.N. Moldagulova and V.V. Serbin, "Gamification and Artificial Intelligence in Education: A Review of Innovative Approaches to Fostering Critical Thinking", *IEEE Access*, Vol. 13, pp. 98699-98728, 2025.

[13] T. Li, Q. Li, M. Zhang, Z. Yuan and Y. Jiang, "Adaptive Streaming Continuous Learning System for Video Analytics", *Proceedings of International Symposium on Quality of Service*, pp. 1-10, 2024.

[14] A.D.S. Sachete, A.V. de Sant'Anna de Freitas Loiola and R.S. Gomes, "AdaptiveGPT: Towards Intelligent Adaptive Learning", *Multimedia Tools and Applications*, Vol. 83, No. 41, pp. 89461-89477, 2024.

[15] F. Ouyang, M. Guo, N. Zhang, X. Bai and P. Jiao, "Comparing the Effects of Instructor Manual Feedback and ChatGPT Intelligent Feedback on Collaborative Programming in Higher Education", *IEEE Transactions on Learning Technologies*, Vol. 17, pp. 2173-2185, 2024.

[16] S. Rathi, S. Mule, S. Nichat, J. Mulla and S. Neware, "Analysis and Implementation of Machine Learning Approaches for Adaptive Learning using Personalized Assessment", *Proceedings of International Conference on Intelligent Data Communication Technologies and Internet of Things*, pp. 1533-1541, 2024.

[17] L. Zhao and H. Wang, "Research on Adaptive Learning System based on Three Core Modules", *Proceedings of International Conference on Information Technology in Medicine and Education*, pp. 447-452, 2019.

[18] Q. Qi, "Application of Reinforcement Learning in Personalized Learning Path Recommendation in Secondary Education Management", *Proceedings of International Symposium on Computer Technology and Information Science*, pp. 38-41, 2025.

[19] Y.J.J. Alawneh, "Adaptive Learning Systems: Revolutionizing Higher Education through AI-Driven Curricula", *Proceedings of International Conference on Knowledge Engineering and Communication Systems*, pp. 1-5, 2024.

[20] A. Ezzaim, "Enabling Sustainable Learning: A Machine Learning Approach for an Eco-Friendly Multi-Factor Adaptive E-Learning System", *Procedia Computer Science*, Vol. 236, pp. 533-540, 2024.

[21] E.G. Rincon-Flores, "Improving the Learning-Teaching Process through Adaptive Learning Strategy", *Smart Learning Environments*, Vol. 11, No. 1, pp. 1-7, 2024.

[22] L. Dong and X. Chen, "Research on Adaptive Learning Model based on Multidimensional Feature Fusion", *Proceedings of International Symposium on Computer Engineering and Intelligent Communications*, pp. 434-438, 2023.

[23] K. Hilali, M. Chergui and A. Ammoumou, "Adaptive Learning Systems: A Comprehensive Overview and Identification of Challenges", *Proceedings of International Conference on Technology Management, Operations and Decisions*, pp. 1-7, 2023.

[24] B. Li, "A Personalized Recommendation Framework based on MOOC System Integrating Deep Learning and Big Data", *Computers and Electrical Engineering*, Vol. 106, pp. 1-8, 2023.

[25] H. Ji, "An Improved Feedback-Feedforward Model-Free Adaptive Iterative Learning Control with High-Order Estimation", *Proceedings of International Conference on Data Driven Control and Learning Systems*, pp. 887-892, 2022.

[26] L. Alfaro, "Work in Progress: Adaptive E-Learning Software Architecture based on Immersive Technologies", *Proceedings of International Conference on World Engineering Education*, pp. 1-8, 2021.

[27] Y. Wang, "A Systematic Review and Meta-Analysis of Self-Determination-Theory-based Interventions in the Education Context", *Learning and Motivation*, Vol. 103, pp. 1-7, 2024.

[28] J. Zhang and F. Qi, "Relationship between Learning Flow and Academic Performance among Students", *Frontiers in Psychology*, Vol. 14, pp. 1-7, 2023.

[29] T. Tan, "Adaptive Moving Average Q-Learning", *Proceedings of International Conference on Knowledge and Information Systems*, pp. 1-6, 2024.

[30] X. Zhao, "Research on Learner Model of Adaptive Learning System", *Proceedings of International Conference on Computer Science and Education*, pp. 906-909, 2021.

[31] H. Yan, "Using Learning Analytics and Adaptive Formative Assessment to Support at-Risk Students in Self-Paced Online Learning", *Proceedings of International Conference on Advanced Learning Technologies*, pp. 396-398, 2020.
[32] O. Ovtšarenko, "Opportunities for Automated E-Learning Path Generation in Adaptive E-Learning Systems", *Proceedings of International Conference on Electrical, Electronic and Information Sciences (eStream)*, pp. 1-4, 2023.
[33] UNESCO, "Technology in Education: A Tool on Whose Terms? Global Education Monitoring Report 2023", Available at https://unesdoc.unesco.org/ark:/48223/pf0000385723, Accessed in 2023.
[34] J. Sweller, "The Development of Cognitive Load Theory", *Educational Psychology Review*, Vol. 35, No. 95, pp. 1-7, 2023.