



Article

A Deterministic State Machine Orchestrator with Local LLM Improving Personalized Education Quality Through Interactive Virtual Tutoring Agent with KPI Tracking

Smail Tigani 

Engineering Unit, Euromed Research Center, Euromed University, Fez 30070, Morocco; s.tigani@ueuromed.org

Abstract

Artificial intelligence is rapidly changing education. However, many learning chatbots are still reactive tools, which respond to arbitrary questions without leading learners through a meaningful pedagogical journey. This article presents a deterministic state-machine orchestrator coupled with a local large language model and a knowledge-graph-framed tutoring strategy for personalized education. The proposed virtual tutoring agent is designed to combine the flexibility of conversational AI with the reliability of explicit instructional states, key performance indicator (KPI) tracking, learner profiling, and controlled transitions between explanation, practice, feedback, assessment, and remediation. The system is not meant to replace the teacher, but rather to act as a teaching co-pilot that provides ongoing feedback, personalized learning paths, accessibility, and safer deployment by processing data locally. The study also presents a compact interview-based evaluation framework and statistical analysis of user perceptions across interactivity, individuality, proactivity, security, accessibility, gamification, and global preference for educational agents over classical chatbots. The findings show that learners appreciate personalized and interactive support and that proactivity is the key feature that distinguishes an educational agent from a regular chatbot. With this article we argue that deterministic orchestration can help make AI tutoring more transparent, controllable, and ethically fit for real learning contexts. Finally, it discusses privacy, educational value, limitations and future improvements to be made before the large-scale adoption of such systems.

Keywords: education technology; quality education; reinforcement learning; agentic artificial intelligence; large language models



Academic Editor: Shuping He

Received: 17 May 2026

Revised: 12 June 2026

Accepted: 23 June 2026

Published: 3 July 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

1. Introduction

Education is moving from simple digitization toward intelligent learning environments that can respond to individual needs in real time. Earlier educational technologies were limited to only transferring existing material to digital platforms. While recent artificial intelligence (AI) systems can generate explanations, evaluate learner responses, and support more flexible interaction. But a chatbot that simply answers prompts is not sufficient for high-quality learning. Education also requires structure, continuity, feedback, and a clear pedagogical intention. For this reason, the present work focuses on an agentic tutoring approach where a local large language model (LLM) is orchestrated by a deterministic state-machine and guided by explicit learning indicators.

The motivation behind this study is practical. In many classrooms, teachers must support learners who progress at different speeds, ask different questions, and need

different forms of feedback. Human tutoring works because it can adapt to the learner. But providing that at scale and continuously is hard. A virtual tutoring agent can help reduce this gap if it is designed as a teaching copilot rather than as an uncontrolled conversational system. Here, the term teaching copilot is used as a generic role descriptor and does not refer to Microsoft Copilot or any other external Copilot software product; therefore, no Copilot software version was used in this study. The proposed approach therefore combines conversational flexibility with deterministic control: the LLM can produce natural explanations, while the orchestrator supervises the pedagogical flow, tracks key performance indicators (KPIs), and decides when the learner should receive clarification, practice, assessment, remediation, or encouragement.

This distinction is important for education. Generative AI can produce fluent text, but fluency is not the same as learning quality, reliability or fairness. The educational agent needs to know where the learner is in the learning process, what concept is being targeted, what misconception might arise and what pedagogically appropriate action to take. In the proposed architecture, the state machine provides this control layer. It constrains the system to transparent instructional states, reduces random behavior, and makes the tutoring process easier to inspect. This is particularly important when AI is applied with students, where privacy, trust, safety, and accountability are just as important as technical performance.

The article presents the design and evaluation of a deterministic state-machine orchestrator connected to a local LLM for personalized tutoring. The problem studied is therefore multi-dimensional: open-ended chatbots often lack pedagogical sequencing, weakly trace learner progress, provide limited personalization beyond the current prompt, raise privacy concerns when learner data are sent to cloud LLMs, and may blur teacher authority if their decisions are not inspectable. The system is positioned as a secure and adaptive teaching copilot able to support individualized progression, interactive feedback and KPI-based monitoring while keeping learning objectives and validation under teacher control. The study also explores how users perceive key dimensions of such a system like interactivity, individuality, proactivity, security, accessibility, and gamification. These dimensions are used to compare the value of an educational agent with that of a conventional chatbot.

The remainder of the paper is organized as follows. Section 2 reviews educational technology, agentic AI, LLM applications, reinforcement learning, and learning analytics. Section 3 presents the materials, evaluation instrument, statistical framework, and deterministic state-machine design. Section 4 reports the statistical and testing results. Section 5 discusses the interpretation of the findings, ethical considerations, privacy, value proposition, and limitations. The final section summarizes the main contribution and outlines future perspectives.

2. Related Works

2.1. Educational Technology

Educational technology has evolved from content delivery platforms toward adaptive environments that can support feedback, assessment, and learner guidance. Recent AI-based systems make it possible to personalize explanations, automate parts of evaluation, and provide immediate assistance to students [1,2]. Nevertheless, the value of these tools depends on their pedagogical grounding. If an AI system is not connected to learning objectives, instructional sequencing, and teacher supervision, it may become an attractive interface without real educational impact [3]. Studies in language and science education show that natural language processing and interactive feedback can help learners correct misconceptions and improve understanding when they are integrated into a coherent learning design [1,4].

2.2. Agentic AI

Agentic AI refers to systems that do more than generate responses: they can monitor a situation, choose actions, and pursue a goal under defined constraints. In education, this ability is useful for reflective practice, tutoring dialogues, and data-driven decision-making [1,3,5]. Pedagogical conversational agents can offer on-demand interaction, but their effect on motivation and learning depends on the quality of the scenario, feedback, and social design [1]. The literature also suggests that agentic systems should not be isolated from teachers. They are more meaningful when used in hybrid human–AI workflows where educators validate content, contextualize recommendations, and preserve learner agency [6]. Transparency, bias, autonomy, and accountability remain central concerns for this type of system [5,7].

2.3. Large Language Models (LLMs) Applications

Large Language Models (LLMs) have opened new possibilities for natural conversation, automated writing support, explanation generation, and personalized content creation [4,8]. Empirical work in EFL learning reports improvements in writing accuracy, speaking fluency, and learner motivation when LLM-based tools are used with clear instructional goals [2]. Other AI models, including transformer-based scoring systems and recommendation engines, can reduce grading time and improve satisfaction with feedback [1]. At the same time, LLMs raise concerns about hallucination, overreliance, plagiarism, and uneven quality of feedback [9]. A responsible approach therefore treats LLMs as assistive components inside a controlled pedagogical process, not as independent replacements for human instruction [1,2].

2.4. Reinforcement Learning Applications

Reinforcement Learning (RL) is relevant to education because it can optimize decisions through feedback from learner interactions [10]. It has been used to personalize interfaces, organize learning content, and recommend adaptive learning pathways [11–13]. Q-learning and deep RL are especially suitable for discrete pedagogical choices, for example selecting the next exercise, adjusting difficulty, or sequencing grammar rules. However, the literature also highlights limitations: many studies rely on simulated learners, short experiments, or small samples [12]. These issues show the need for practical architectures that combine adaptive decision-making with explicit pedagogical control.

2.5. Learning Analytics

Learning analytics (LA) uses educational data to support instruction, identify at-risk learners, and personalize feedback [1]. Modern LA systems increasingly analyze behavioral, linguistic, and affective signals such as hesitation, confusion, curiosity, and stress in order to support real-time pedagogical decisions [5]. In language education, LA models can map progression using frameworks such as Bloom’s taxonomy and can optimize content delivery through interaction logs [3]. Despite these advances, many tools still stop at dashboards or predictions and do not translate insights into concrete teaching actions [14]. The broader engineering literature also supports this need for explicit orchestration. Agile software processes emphasize iterative validation and stakeholder feedback [15]. Graph mining and document-mining methods show how structured representations can support scalable knowledge organization [16,17]. Graph neural and code-generation studies illustrate the value of feature augmentation and domain-specific LLM assistance [18,19]. Fine-tuning and path-planning studies further show how controlled models can support adaptive technical workflows [20,21]. Markov-decision, portfolio-optimization, and security-verification studies provide additional evidence for structured decision-making un-

der constraints [22–24]. Graph-mining and probabilistic-modeling research also contributes methods for representing complex relations and uncertainty [25,26]. Finally, agentic-AIOps, intelligent-recognition, and sustainable process-automation studies show how orchestrated AI workflows can be transferred to applied decision environments [27–29]. Human-centered LA, co-designed with educators, is therefore essential for ethical, equitable, and useful deployment [6,7].

2.6. Gap Analysis

The reviewed literature confirms strong progress in AI-supported education, but it also reveals a gap between conversational capability and pedagogical control. Existing systems often focus on one dimension, such as adaptive feedback, analytics, or language generation. Fewer works combine these dimensions inside a transparent orchestration model that can guide an LLM through predefined learning states while tracking KPIs and preserving privacy through local deployment. The proposed system addresses this gap by connecting an educational agent to a deterministic state machine, allowing the platform to remain conversational while keeping the learning pathway explicit, inspectable, and aligned with teacher-defined objectives.

Compared with representative studies, the proposed approach retains the pedagogical orientation and personalized-learning objectives already present in prior work while extending them through explicit orchestration and local LLM integration. Yusuf et al. [1], Schorcht et al. [3], and Diebel et al. [5] all address pedagogical use and personalized learning, but they do not combine these aspects with an LLM-driven tutoring agent. Learning analytics is considered in Yusuf et al. [1] and Diebel et al. [5], whereas Schorcht et al. [3] gives less emphasis to this dimension. In contrast, the present system integrates pedagogical support, personalized learning, learning analytics, accessibility, and LLM-based interaction within a single deterministic state-machine framework. This combination distinguishes the proposed tutor from systems that either remain non-generative, rely mainly on analytics, or lack an accessible conversational interface.

3. Methodology

This section presents the methodological foundations used to design, implement, and evaluate the proposed proactive educational tutoring agent. It first describes the implementation infrastructure, including the local text-generation and speech-generation models, the Next.js 14 platform, and the MongoDB 8 course-content database. It then details the service-oriented system architecture that connects the web platform to the AI services through REST APIs. The evaluation methodology is introduced through the interview instrument, which compares a generic QA chatbot with the proposed proactive tutor and measures satisfaction across key educational-AI features. Finally, the section formalizes the deterministic orchestration mechanism as a state machine, defining the states, actions, and transition rules that coordinate tutoring behaviors such as introduction, explanation, question answering, evaluation, and session closure.

3.1. Implementation Infrastructure

All experiments were conducted on a standard laptop equipped with an Intel Core i7 CPU, 16 GB of RAM, and a 1 TB SSD. The system hosted the PyTorch 2.12 Qwen2.5-1.5B-Instruct model for text-to-text tasks and PyPI Supertonic 3 for text-to-speech tasks. The platform was developed using Next.js 14 and MongoDB 8 as the NoSQL database for hosting the course content that drives the teaching agent.

3.2. System Architecture

The proposed platform follows a Service-Oriented Architecture (SOA) in which the web application is separated from the AI inference components and communicates with them through REST APIs. The web platform acts as the main orchestration and interaction layer: it manages learner sessions, course content, KPI tracking, state-machine transitions, and user-interface events. When the tutor requires natural-language output, the platform sends a REST request to the TextGeneration service, which produces pedagogical explanations, answers, or feedback according to the current learning context. When spoken interaction is required, the generated text is forwarded through another REST endpoint to the Speech Generation Service, which converts the tutor response into audio for avatar-based delivery. This SOA design decouples the user-facing platform from the specialized AI services, making the system easier to maintain, scale, and replace. The text-generation and speech-generation services can be deployed, updated, or optimized independently without changing the core web platform. REST APIs also provide a clear communication contract between components, allowing the deterministic orchestrator to call each service only when the current pedagogical state requires it. Figure 1 summarizes this architecture and shows how the web platform connects to the TextGeneration service and the Speech Generation Service.

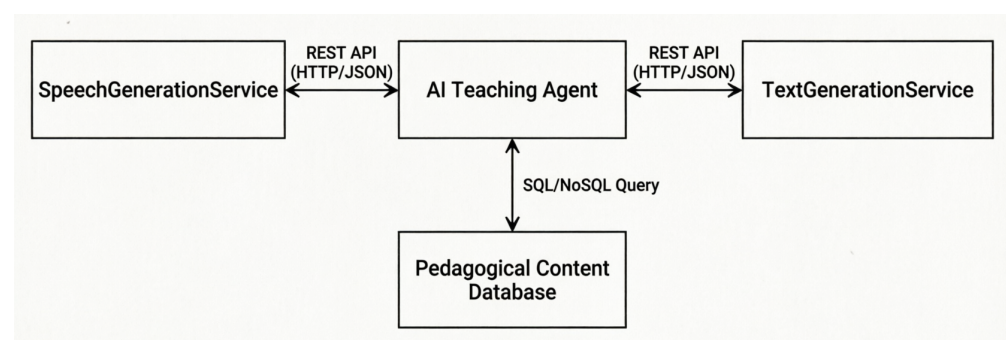


Figure 1. Service-Oriented Architecture connecting the web platform to the TextGeneration service and Speech Generation Service through REST APIs.

3.3. Interview Instrument Method

To evaluate user preference for our proactive educational chatbot (Assistant B) versus a generic QA system (Assistant A), we developed a three-question instrument balancing quantitative rigor with participant efficiency. The comparative preference question used three response values: Prefer A, Neutral, and Prefer B, where Assistant A represents a generic QA chatbot and Assistant B represents the proactive tutor. The feature satisfaction question used a five-star scale from 1 = Not satisfying to 5 = Extremely satisfying across Accessibility, Traceability, Individuality, Proactivity, Interactivity, Security, and Gamification.

The evaluation involved 64 volunteer participants recruited through convenience sampling from the university community and professional contacts familiar with digital learning tools. The sample included students and employed participants distributed across the age categories reported in Table 1. Participants used a short structured course module prepared in the platform, consisting of slide-based instructional content followed by slide-level questions. Each participant was exposed to both Assistant A and Assistant B on the same learning material: Assistant A provided reactive question answering, whereas Assistant B followed the deterministic tutoring flow with introduction, explanation, waiting, answering, evaluation, and closure behaviors. The interaction lasted approximately one short learning session per participant. To reduce order effects, the demonstration order was alternated when possible between Assistant A first and Assistant B first; however, because this was a prototype-oriented study rather than a controlled classroom trial,

the exposure order was not fully randomized. The results are therefore interpreted as user-perception evidence rather than proof of learning effectiveness.

Table 1. Interview Instrument Questions and Response Values.

Question	Values
Participant Age	10–20; 21–30; 31–40; 41–50
Participant Gender	Female; Male
Participant Status	Student; Employed
Comparative Preference	Prefer A; Neutral; Prefer B
Feature Satisfaction Rating	1; 2; 3; 4; 5

This evaluation design was selected to verify feasibility and perceived usefulness before deploying the system in real classrooms. The next validation stage will use a longitudinal classroom protocol with pre-test, post-test, and delayed retention measures, together with teacher observations and platform logs. This future protocol will also broaden recruitment beyond the initial university and professional-contact sample by including different educational levels, age groups, disciplines, and digital-literacy profiles. In addition, teachers will be involved as co-designers of the KPI definitions, state-transition rules, and content-quality checks so that the deterministic flow remains pedagogically meaningful rather than only technically consistent.

3.4. Statistical Analysis Methods

We employ a mixed analytical approach combining descriptive statistics, inferential hypothesis testing, and correlation analysis to address two research questions: (1) Is there significant preference for the proactive approach? (2) Which features drive user satisfaction?

3.4.1. Mann-Whitney U Test for Comparative Preference Analysis

The Mann-Whitney U test, also known as the Wilcoxon rank-sum test, was used as a non-parametric procedure for comparing two independent response groups. In this study, the test supports the analysis of participants' comparative preference between Assistant A (Generic QA) and Assistant B (Proactive Tutor). The sample sizes are denoted consistently as n_A for Assistant A and n_B for Assistant B. This choice is appropriate because the preference variable is ordinal/categorical and does not require the normality assumption imposed by an independent-samples t -test. The test evaluates the null hypothesis (H_0) that the two preference distributions are equivalent. Equivalently, the probability that a randomly selected observation from one group exceeds a randomly selected observation from the other group is the same in both directions. The alternative hypothesis (H_1) states that the two distributions differ, indicating that one system tends to receive higher ranked preference than the other.

$$H_0 : P(X_A > X_B) = P(X_B > X_A) \quad \text{vs.} \quad H_1 : P(X_A > X_B) \neq P(X_B > X_A) \quad (1)$$

where X_A and X_B represent the ranked preference observations for Assistant A and Assistant B, respectively. To compute the statistic, all observations from both groups (n_A and n_B) are pooled and ranked from smallest to largest. Let R_A and R_B denote the rank sums for Assistant A and Assistant B, respectively. The two U statistics are calculated as follows:

$$U_A = n_A n_B + \frac{n_A(n_A + 1)}{2} - R_A \quad (2)$$

$$U_B = n_A n_B + \frac{n_B(n_B + 1)}{2} - R_B \quad (3)$$

The final test statistic U is the smaller of U_A and U_B for two-sided comparisons, with $U_A + U_B = n_A n_B$. In discrete survey data, tied ranks are common; therefore, the variance should be adjusted for ties when computing significance. For sufficiently large samples, the distribution of U can be approximated by a normal distribution, allowing the calculation of a Z-score:

$$Z = \frac{U - \mu_U}{\sigma_U} \quad (4)$$

where $\mu_U = \frac{n_A n_B}{2}$ and σ_U is the standard deviation adjusted for ties. The resulting p -value determines statistical significance. If $p < \alpha$ (typically 0.05), the null hypothesis is rejected, indicating that the ranked preference distributions differ significantly. To report the magnitude of the Mann-Whitney result, we also calculate the rank-biserial correlation (r_{rb}). The statistic r_{rb} is a non-parametric effect size that expresses the directional separation between two ranked groups. Values range from -1 to $+1$, where 0 indicates no rank advantage, positive values indicate that the target group tends to receive higher ranks, and values close to $+1$ indicate very strong separation in favor of the target group. For the directional comparison favoring Assistant B, it is computed as:

$$r_{rb} = \frac{2U_B}{n_A n_B} - 1 \quad (5)$$

where U_B is the Mann-Whitney statistic in favor of Assistant B, n_A is the number of participants preferring Assistant A, and n_B is the number of participants preferring Assistant B. In the context of this study, the Mann-Whitney U test provides a coherent way to test whether participants' coded preferences favor the proactive tutor over the generic question-answering assistant. The procedure is retained because it is robust for non-normal and ordinal survey responses, while the complementary descriptive statistics report the magnitude of the observed preference in interpretable percentages.

3.4.2. Feature Pearson Correlation Analysis

Pearson's product-moment correlation coefficient was used to examine the linear association between pairs of perceived AI feature ratings. While the Mann-Whitney U test evaluates the comparative preference between the two assistant types, Pearson correlation evaluates how strongly the feature-level evaluations move together across participants. In this study, the method is applied to the seven Likert-coded educational AI features: accessibility, traceability, individuality, proactivity, interactivity, security, and gamification. The analysis helps identify whether participants who assign high value to one design dimension also tend to assign high value to another dimension. For two feature variables X and Y , Pearson's correlation coefficient r_{XY} is defined as the standardized covariance between the variables:

$$r_{XY} = \frac{\text{cov}(X, Y)}{s_X s_Y} \quad (6)$$

where $\text{cov}(X, Y)$ is the sample covariance, and s_X and s_Y are the sample standard deviations of the two feature variables. Equivalently, for n participants, the coefficient is computed as:

$$r_{XY} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (7)$$

where x_i and y_i are the observed ratings for participant i , and $\bar{x}$ and $\bar{y}$ are the corresponding sample means. The coefficient ranges from -1 to $+1$. A value close to $+1$ indicates a strong positive linear relationship, meaning that higher ratings on one feature are associated with higher ratings on the other. A value close to -1 indicates a strong negative linear relationship, meaning that higher ratings on one feature are associated with lower ratings

on the other. A value near 0 indicates little or no linear association. Statistical significance is assessed using the t transformation of the correlation coefficient:

$$t = r \sqrt{\frac{n-2}{1-r^2}} \quad (8)$$

with $df = n - 2$ degrees of freedom. Correlations are interpreted using both magnitude and significance: weak correlations suggest limited co-movement between features, moderate correlations indicate meaningful feature alignment, and strong correlations suggest that the paired features may represent a shared latent perception of educational AI value. Because the feature ratings are measured on the same 5-point response scale, the resulting correlation matrix provides a compact diagnostic view of the perceived design structure underlying the agentic tutoring system.

3.5. Agent Orchestration State-Machine Design

The orchestration layer can be formalized as a deterministic finite state machine whose control variable is the pair (agentState, agentAction). In the current implementation, agentState is fixed to ACTIVITY during protocol execution, while agentAction selects the protocol to run. The Boolean lock isAgentBusyRef.current acts as an execution guard: if the lock is active, no transition is fired; otherwise, the selected protocol is executed exactly once and the lock is released in the finally block. Let the orchestrator be defined by

$$\mathcal{M}_{\text{orchestration}} = \langle \mathcal{S}, \mathcal{A}, \delta, s_{\text{wait}}, s_{\text{close}} \rangle \quad (9)$$

where $\mathcal{S}$ is the set of protocol states, $\mathcal{A}$ is the set of executable actions, δ is the transition function, s_{wait} is the initial waiting state, and s_{close} is the final state reached when the session is closed.

The six-state structure was selected because it represents the minimum controllable pedagogical cycle needed by the prototype. WAIT prevents uncontrolled generation and gives the learner or interface time to trigger the next action. INTRODUCE establishes the session goal and course context. EXPLAIN delivers teacher-defined content before free conversation begins. ANSWER handles learner questions without losing the current instructional position. EVALUATE checks assimilation through explicit questions and KPI updates. CLOSE terminates the session and preserves the trace of the interaction. This design was preferred over a purely prompt-based chatbot because prompts alone do not guarantee sequencing or traceability; over a fixed tutoring script because the local LLM can still adapt explanations and answers to learner input; and over reinforcement learning because the current prototype requires transparent teacher-inspectable rules before collecting enough longitudinal data for policy learning.

3.5.1. States Set

The state s_{wait} corresponds to WAIT and executes waitProtocol() while the tutor waits for learner input or an external event. The state $s_{\text{introduce}}$ corresponds to INTRODUCE and executes introduceProtocol(), which opens the pedagogical interaction after initSession(). The state s_{explain} corresponds to EXPLAIN and executes explainProtocol() to present learning content. The state s_{answer} corresponds to ANSWER and executes answerProtocol() when the learner asks a question. The state s_{evaluate} corresponds to EVALUATE and executes evaluateProtocol() to assess learner performance. The state s_{close} corresponds to CLOSE and executes closeProtocol() before disconnecting the avatar. Formally, the state set $\mathcal{S}$ is defined as the following:

$$\mathcal{S} = \{s_{\text{wait}}, s_{\text{introduce}}, s_{\text{explain}}, s_{\text{answer}}, s_{\text{evaluate}}, s_{\text{close}}\} \quad (10)$$

3.5.2. Actions Set

The action a_{start} is produced by `initSession()`, which sets `sessionStarted` to true, resets the KPI counters, and sets `agentAction` to `INTRODUCE`. The actions $a_{\text{introduce}}$, a_{wait} , a_{explain} , a_{answer} , a_{evaluate} , and a_{close} denote the execution of their corresponding protocols. Formally, the action set $\mathcal{A}$ is given by:

$$\mathcal{A} = \{a_{\text{start}}, a_{\text{introduce}}, a_{\text{wait}}, a_{\text{explain}}, a_{\text{answer}}, a_{\text{evaluate}}, a_{\text{close}}\} \quad (11)$$

The pedagogical layer updates concrete KPIs during these actions. The main indicators are slide-completion ratio, answer correctness, number of learner questions, repeated-question frequency, time spent on a slide, number of hints requested, and evaluation-attempt count. For example, a correct answer after explanation increases the assimilation score and allows the transition from `EVALUATE` to `WAIT` or to the next slide; an incorrect answer or low confidence score triggers additional explanation before a new evaluation attempt; repeated questions on the same concept indicate possible misconception and move the agent from `WAIT` to `ANSWER` or `EXPLAIN`; long inactivity keeps the agent in `WAIT` or produces a light prompt for engagement; and session completion or learner exit triggers `CLOSE`. In this way, state transitions are linked to observable learner behavior rather than to arbitrary dialogue turns.

3.5.3. Transition Function

The deterministic transition function is $\delta : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$. Table 2 defines the state machine in a comprehensible form by pairing each transition rule with its implementation meaning.

Table 2. Deterministic transition function of the orchestration state machine.

Transition	Short Meaning
$\delta(s_{\text{wait}}, a_{\text{start}}) = s_{\text{introduce}}$	Start the tutor session.
$\delta(s_{\text{introduce}}, a_{\text{introduce}}) = s_{\text{wait}}$	Return after the greeting.
$\delta(s_{\text{wait}}, a_{\text{explain}}) = s_{\text{explain}}$	Open an explanation.
$\delta(s_{\text{wait}}, a_{\text{answer}}) = s_{\text{answer}}$	Answer a learner question.
$\delta(s_{\text{explain}}, a_{\text{wait}}) = s_{\text{wait}}$	End the explanation.
$\delta(s_{\text{answer}}, a_{\text{wait}}) = s_{\text{wait}}$	End the answer.
$\delta(s_{\text{wait}}, a_{\text{evaluate}}) = s_{\text{evaluate}}$	Start assessment.
$\delta(s_{\text{evaluate}}, a_{\text{wait}}) = s_{\text{wait}}$	End assessment.
$\delta(s, a_{\text{close}}) = s_{\text{close}}$	Close the session.

Thus, the executor fires only when a valid state–action pair is present and no protocol is already running. The final state is s_{close} , which is one state in $\mathcal{S}$ and is reached when the session is closed.

3.5.4. State Machine

Figure 2 shows the deterministic orchestration graph. The start transition leaves s_{wait} and enters $s_{\text{introduce}}$, pedagogical actions cycle through the waiting state, and the close action can interrupt any active pedagogical state to terminate the avatar session in s_{close} .

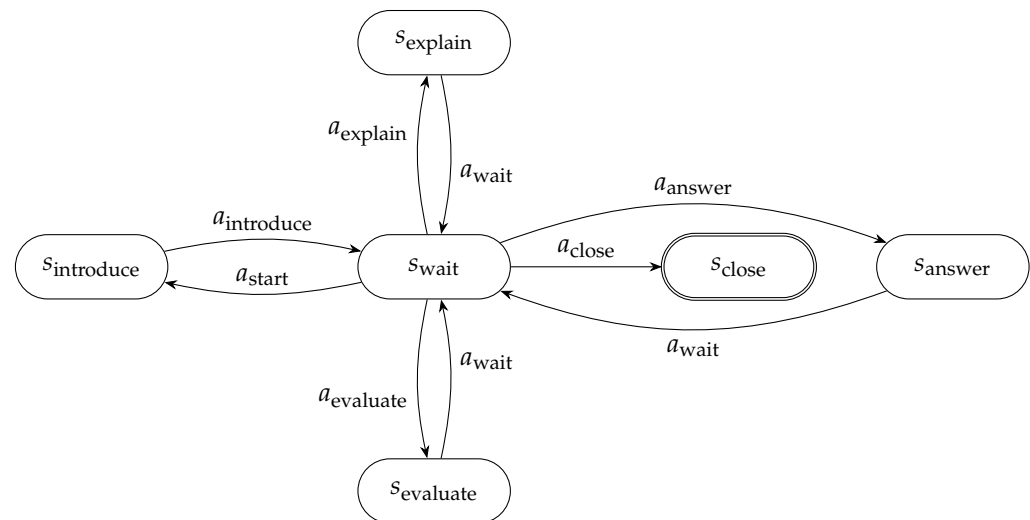


Figure 2. Deterministic state machine corresponding to the Next.js orchestration code. Edges represent normal protocol execution.

4. Results

4.1. Statistical Analysis Results

This section presents the quantitative analysis of $N = 64$ participant responses regarding preferences for Educational Agents versus Educational Chatbots. The analysis covers descriptive statistics, scale reliability, group comparisons, and correlational structures.

4.1.1. Descriptive Statistics

Participants rated seven key features of AI educational systems on a 5-point Likert scale (1 = Strongly Disagree, 5 = Strongly Agree). Table 3 summarizes the central tendency and dispersion for each feature.

Table 3. Descriptive Statistics for Educational AI Features ($N = 64$).

Feature	Mean ($\bar{x}$)	SD (s)	Median	Mode	Min	Max
Accessibility	4.20	1.01	5	5	1	5
Traceability	3.97	1.04	4	5	1	5
Individuality	4.11	1.07	4	5	1	5
Proactivity	3.80	1.17	4	5	1	5
Interactivity	4.27	0.95	5	5	1	5
Security	4.23	1.26	5	5	1	5
Gamification	3.94	1.10	4	5	1	5

Figure 3 indicates a generally positive reception of AI features across the sample. Interactivity ($\bar{x} = 4.27$) and Security ($\bar{x} = 4.23$) received the highest average ratings, suggesting that users prioritize engaging interfaces and data safety. Proactivity received the lowest mean score ($\bar{x} = 3.80$) with the highest standard deviation ($s = 1.17$), indicating greater variance in user comfort levels regarding bot-initiated guidance compared to reactive features.

4.1.2. Mann-Whitney Test Results for Group Preference Comparison

For Question 1, participants selected their preferred system after comparing Assistant A (Generic QA) with Assistant B (Proactive Tutor). Responses were coded as Assistant A = 0 and Assistant B = 1. The observed distribution was $n_A = 21$ for the generic Assistant/Chatbot and $n_B = 43$ for the proactive Educational Agent. Because this response is categorical and non-normally distributed, the Mann-Whitney U test was applied only

to this comparative-preference question. The null hypothesis (H_0) states that the ranked preference distribution does not favor Assistant B over Assistant A, while the alternative hypothesis (H_1) states that participants show a stronger preference for Assistant B.

$$H_0 : P(X_B > X_A) = 0.5 \quad (12)$$

$$H_1 : P(X_B > X_A) > 0.5 \quad (13)$$

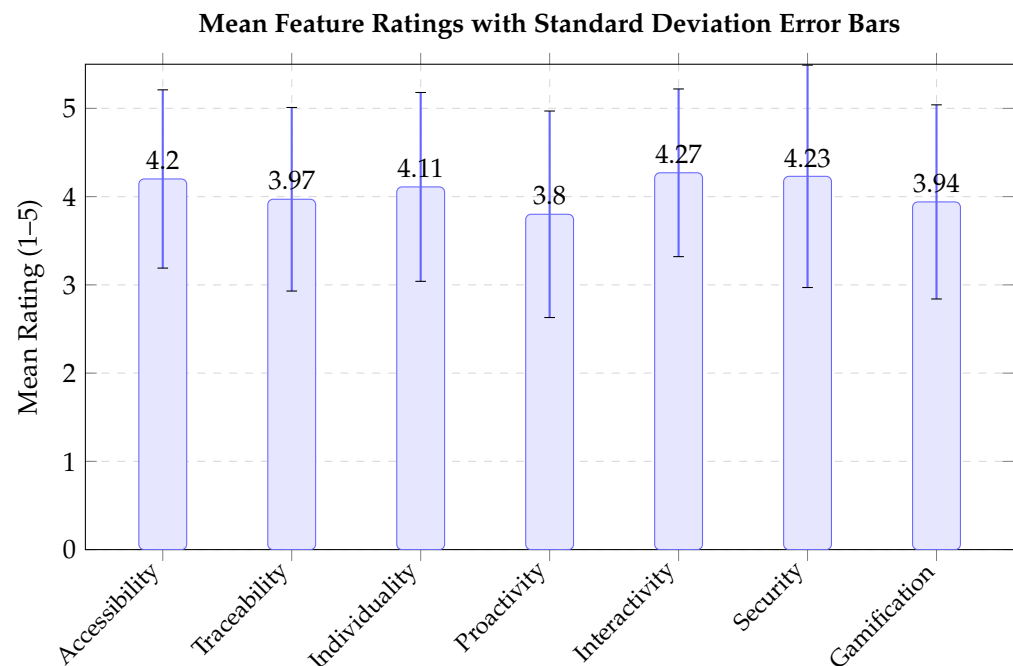


Figure 3. Mean ratings for seven AI educational features. Error bars represent ± 1 standard deviation.

Table 4 presents the Mann-Whitney U statistic for Question 1 only, preserving the results as a preference-level analysis rather than a feature-level comparison.

Table 4. Mann-Whitney U Test for Question 1 Comparative Preference.

Question	Assistant A (n_A)	Assistant B (n_B)	U_B	p -Value	r_{rb}
Comparative Preference	$n_A = 21$ (32.8%)	$n_B = 43$ (67.2%)	903.0	<0.001	1.00

The rank-biserial effect size (r_{rb}) was calculated to quantify the magnitude and direction of the preference difference. In this context, r_{rb} indicates how strongly the ranked binary preference outcome favors the proactive tutor over the generic assistant. The value was calculated as follows:

$$r_{rb} = \frac{2(903.0)}{21 \times 43} - 1 = 1.00 \quad (14)$$

The Mann-Whitney result for Question 1 shows a statistically significant preference for Assistant B (Proactive Tutor) over Assistant A (Generic QA) ($U_B = 903.0$, one-tailed $p < 0.001$). In practical terms, 67.2% of participants selected the proactive agentic tutor, whereas 32.8% selected the generic assistant. The rank-biserial correlation ($r_{rb} = 1.00$) indicates a maximal positive effect in the coded comparison, meaning that the ranked preference outcome is completely separated in favor of Assistant B under the binary coding used for Question 1. Because the outcome is binary, this result was also checked using a simpler preference-proportion interpretation: under a null proportion of 0.50, the observed 43 of 64 preferences for Assistant B gives an estimated preference proportion

of $\hat{p} = 0.672$ with an approximate 95% confidence interval of 0.557–0.787. This interval is above the indifference point of 0.50, supporting the robustness of the descriptive conclusion while avoiding overstatement. Therefore, the finding is interpreted as evidence of a clear aggregate user preference for the agentic tutor, while the feature ratings and correlations reported in the following sections explain which design dimensions are associated with that preference.

4.1.3. Feature Correlation Analysis Result

Table 5 presents the Pearson feature-correlation matrix for the seven educational AI design dimensions. The matrix shows that all observed pairwise associations are positive, indicating that participants who valued one feature generally tended to value the other features as well. This pattern supports the interpretation that the proposed agentic tutor is evaluated as an integrated educational experience rather than as a collection of isolated functions.

Table 5. Pearson Correlation Coefficients Between AI Features ($N = 64$).

	Acc	Tra	Ind	Pro	Int	Sec	Gam
Accessibility	1.00						
Traceability	0.75	1.00					
Individuality	0.64	0.66	1.00				
Proactivity	0.37	0.47	0.55	1.00			
Interactivity	0.41	0.36	0.68	0.45	1.00		
Security	0.47	0.35	0.52	0.53	0.36	1.00	
Gamification	0.31	0.33	0.37	0.51	0.32	0.59	1.00

To make these relationships easier to inspect visually, Figure 4 converts the same correlation matrix into a boxed heatmap. The chart preserves the feature order used in Table 5, places identical labels on both axes, and uses darker blue shading to highlight stronger positive associations between feature pairs.

The heatmap reinforces the Pearson correlation table by showing a uniformly positive association structure among the seven educational AI features. The strongest relationships appear between Accessibility and Traceability ($r = 0.75$), Individuality and Interactivity ($r = 0.68$), and Traceability and Individuality ($r = 0.66$), suggesting that participants perceived transparent, personalized, and interactive support as mutually reinforcing qualities. Moderate positive correlations involving Proactivity, especially with Individuality ($r = 0.55$), Security ($r = 0.53$), and Gamification ($r = 0.51$), indicate that proactive tutor behavior was not evaluated in isolation but was linked to broader expectations of adaptive, safe, and engaging learning assistance. The weakest correlations, such as Accessibility with Gamification ($r = 0.31$) and Interactivity with Gamification ($r = 0.32$), remain positive but comparatively limited, implying that gamified elements contribute less consistently to the shared perception of educational AI quality than personalization, transparency, and interaction.

4.2. Teaching Copilot Testing Results

To validate the practical feasibility of the proposed approach, we developed a web application that implements the deterministic state-machine orchestrator described in this paper. The prototype assumes that the course content has already been prepared and structured in a separate authoring space. Once the learner logs into the learning environment and selects a course, the teaching agent is loaded automatically with the corresponding course context, slides, and evaluation items.

Pearson Feature Correlation Heatmap ($N = 64$)

	Accessibility	Traceability	Individuality	Proactivity	Interactivity	Security	Gamification
Accessibility	1.00	0.75	0.64	0.37	0.41	0.47	0.31
Traceability	0.75	1.00	0.66	0.47	0.36	0.35	0.33
Individuality	0.64	0.66	1.00	0.55	0.68	0.52	0.37
Proactivity	0.37	0.47	0.55	1.00	0.45	0.53	0.51
Interactivity	0.41	0.36	0.68	0.45	1.00	0.36	0.32
Security	0.47	0.35	0.52	0.53	0.36	1.00	0.59
Gamification	0.31	0.33	0.37	0.51	0.32	0.59	1.00

Darker cells indicate stronger positive Pearson correlations.

Figure 4. Boxed heatmap of Pearson correlations between the seven AI educational features. Each cell reports the correlation coefficient, with darker blue cells indicating stronger positive associations.

The agent is represented by an interactive avatar controlled by the state machine. During a learning session, the avatar guides the learner through the slides one by one by following a deterministic instructional sequence. It begins with an introduction process, continues with an explanation process for the current slide, and then enters a waiting process in which it listens for learner interaction. If the learner asks a question, the system activates speech recognition, converts the learner's speech into text, and sends the resulting prompt to the local LLM together with the relevant course context. As the local LLM generates the answer incrementally, the response chunks are transmitted to the avatar-streaming component, which pronounces the generated content and maintains the conversational flow.

After each slide, the application loads evaluation questions and their corresponding answer options in order to assess the learner's assimilation of the slide content. This testing workflow demonstrates that the proposed architecture can combine structured pedagogical control, local LLM-based question answering, speech interaction, avatar-based delivery, and slide-level assessment within a unified teaching copilot environment.

To complement the aggregate survey results, three users who interacted with the platform were asked to provide an exploratory expert-style assessment of the same tested dimensions. Each assessor rated the implemented platform on a 5-point scale, where 1 indicates a weak presence of the dimension and 5 indicates a very strong presence. Figure 5 visualizes these three individual profiles across Accessibility, Traceability, Individuality, Proactivity, Interactivity, Security, and Gamification.

The radar profiles show that the three assessors consistently rated the platform strongly on Interactivity, Security, Individuality, and Accessibility. The comparatively smaller radial values for Proactivity and Gamification are consistent with the aggregate descriptive statistics and indicate dimensions where future iterations can further improve proactive pedagogical intervention and motivational game-like elements.

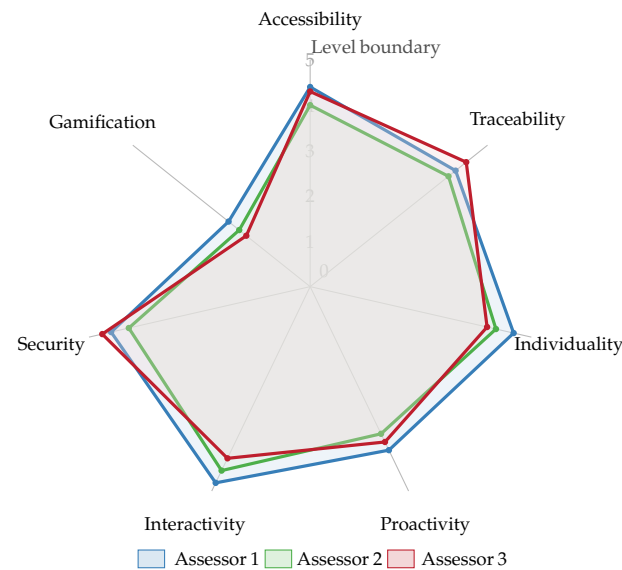


Figure 5. Radar chart of three platform-user assessments across the seven tested educational AI dimensions. Ratings use a 1–5 scale, with larger polygons indicating stronger perceived support for a dimension.

5. Discussion

5.1. Interpretation of Statistical Findings

The results indicate that users do not evaluate educational AI only by its ability to answer questions. They also value the way the system interacts, adapts, protects data, and supports progression. The statistical analysis confirms that while users broadly appreciate several AI features, Proactivity is a key differentiator between an educational agent and a conventional chatbot. The observed effect size and significance support the idea that learners prefer systems that can guide them, not only respond to them.

This interpretation must remain cautious. The 67.2% preference for Assistant B is a positive user-perception result, not a direct measure of educational effectiveness. The present evaluation does not yet establish achievement gains, retention, grade improvement, reduced teacher workload, or long-term classroom impact. It shows that participants perceived the proactive tutor as more valuable than the generic QA assistant under the prototype demonstration conditions. Future controlled studies should therefore connect the state-machine KPIs to learning outcomes such as pre/post-test gains, delayed retention tests, completion time, teacher feedback time, and classroom engagement.

The high ratings for Interactivity, Individuality, Security, and Accessibility suggest that participants expect an AI tutor to be responsive, personal, safe, and easy to use. These dimensions are not independent from the pedagogical design: an interactive system must still follow a meaningful learning sequence, and a personalized system must still remain explainable to teachers and learners. The deterministic state-machine design responds to this requirement by giving the virtual tutor a controlled structure for moving from explanation to practice, feedback, assessment, and remediation.

Gamification received comparatively lower values than the core tutoring dimensions. This does not mean that game elements are useless, but it suggests that learners may first expect clarity, relevance, and useful feedback. In this context, gamification should be used carefully as a support for motivation, not as a substitute for pedagogical quality. The radar analysis also indicates that proactivity can still be improved, particularly by making interventions feel helpful rather than intrusive.

5.2. Ethical Considerations and Privacy

AI systems in education must be designed with strong ethical safeguards because they can influence learning paths, confidence, assessment, and teacher decisions. The first ethical requirement is transparency: learners and teachers should know when they are interacting with an AI system, what data are being used, and how recommendations are produced. The deterministic state-machine layer contributes to this transparency. It provides a fuller and easier to audit design of the AI's pedagogical state.

Privacy is also central. Educational data may include performance records, mistakes, learning habits, and sometimes sensitive personal information. A local LLM architecture can reduce exposure by limiting the transfer of learner data to external services. However, local deployment alone is not enough. The platform should apply data minimization, role-based access, secure storage, anonymized analytics where possible, and clear consent procedures. Learners should also be able to review, correct, or delete personal data according to institutional and legal requirements.

For the prototype evaluation reported in this article, participation was voluntary and responses were analyzed in aggregate form. No personally identifying information was required for the statistical analysis, and the reported dataset was anonymized before interpretation. Participants were informed that the activity concerned a prototype educational AI system and that their responses would be used for research reporting. The authors' roles were separated as far as possible within the constraints of a small prototype study: development focused on the platform, local LLM integration, state-machine orchestration, and avatar interaction, whereas evaluation focused on collecting anonymous user perceptions and interpreting aggregate results. Future classroom deployments should obtain formal institutional ethics approval when required and should define a stronger separation between system developers, teachers, and evaluators.

Another ethical issue is the risk of overdependence. If students rely too much on the agent, they may become less willing to struggle productively, verify information, or develop independent reasoning. For this reason, the agent should encourage reflection, ask guiding questions, and explain reasoning instead of simply giving final answers. Teachers should remain responsible for educational decisions, while the agent provides support, monitoring, and additional practice opportunities.

5.3. Value Proposition

The main value of the proposed system is the combination of personalization and control. For learners, it can provide immediate feedback, adaptive explanations, and continuous access to practice. For teachers, it can act as a copilot that tracks KPIs, highlights learner difficulties, and reduces repetitive support tasks. For institutions, the architecture offers a pathway toward scalable tutoring while keeping attention on security, accessibility, and quality assurance.

Compared with a generic chatbot, the deterministic orchestrator makes the learning process more predictable. This is valuable because educational applications require more than creative answers; they require consistent behavior, traceable decisions, and alignment with objectives. The system can therefore support quality education by helping learners receive the right type of assistance at the right moment, while allowing teachers to understand and adjust the pedagogical flow.

5.4. Limitations

This work has several limitations. First, the evaluation relies mainly on user perceptions and a compact interview framework. Such data are useful for early validation, but they do not replace long-term classroom studies measuring learning gains, retention,

motivation, teacher workload, and classroom engagement over several learning units or semesters. Second, although the prototype was tested with students and employed participants, the number and diversity of assessors remain limited. Future experiments should expand the sample to include learners from different ages, school levels, disciplines, languages, socioeconomic contexts, and digital-literacy levels.

Third, the quality of the system still depends on the quality of the instructional content, prompts, KPI definitions, and state transitions. A deterministic state machine can improve control, but it cannot automatically guarantee pedagogical excellence if these elements are poorly designed or not updated with teacher feedback. The current version therefore requires human validation of learning objectives, remediation paths, and threshold values before classroom use. Finally, local LLM deployment improves privacy and autonomy, but it may introduce constraints related to hardware availability, inference speed, model size, update frequency, multilingual performance, and response quality. These constraints are important for institutions with limited computing resources or rapidly changing course content.

Future work should therefore combine larger empirical studies, teacher-centered co-design, stronger privacy audits, and comparisons with alternative adaptive tutoring architectures. A concrete extension will be a teacher-facing authoring and analytics interface where instructors can collaboratively define states, adjust KPI thresholds, review problematic transitions, annotate learner difficulties, and use feedback from classroom practice to optimize the tutoring flow. This will help transform the proposed system from a technically controlled prototype into a pedagogically governed educational tool.

6. Conclusions and Perspectives

This article proposed a deterministic state-machine orchestrator with a local LLM for a personalized virtual tutoring agent in education. The main idea is that educational AI should not be limited to open-ended conversation. It should also include pedagogical structure, KPI tracking, secure data handling and transparent control over the learner's progression. By combining a conversational model with explicit instructional states, the proposed system aims to offer the flexibility of LLMs while reducing the unpredictability that often limits their use in formal learning environments.

The results show that users appreciate interactive, individualized, secure, and accessible tutoring support. They also indicate that proactivity is the feature that distinguishes an educational agent from a traditional chatbot. This supports the design choice of an agent that can guide the learner through explanation, practice, feedback, assessment, and remediation instead of waiting passively for questions. Additionally, the findings suggest that gamification should be added carefully and should serve learning objectives rather than dominate the educational experience.

From a practical perspective, the proposed architecture provides clear benefits for learners, teachers, and institutions. Learners get immediate and personalized support. Teachers receive help in monitoring challenges and reinforcing their instruction. Institutions obtain a more controllable and privacy-aware model for AI-assisted education. Future work will focus on larger-scale classroom validation, richer behavioral analytics, stronger privacy evaluation, teacher-facing dashboards, and refinement of the proactive intervention rules. Specifically, the next phase of research will include longitudinal classroom experiments with varied learner groups, comparing learning results with those from traditional digital-learning support, and developing collaborative tools for teachers to better manage key performance indicators, state definitions, and feedback loops. These steps are necessary to move from a promising prototype toward a reliable educational technology that can contribute to quality, inclusive, and human-centered learning.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/bdcc10070219/s1>. Figure (A) illustrates the Web UI of the interactive teaching copilot, featuring a speech-based mode (displaying spectrograms). Additionally, Table (A) summarizes the survey data from our Agentic AI for Education study.

Funding: The author gratefully acknowledges Euromed University of Fez for its support.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to the anonymous, minimal-risk nature of the study. Under Moroccan Law 09-08 on the protection of individuals with regard to the processing of personal data, regulations apply only to identifiable data. Our dataset is anonymous by design, it falls outside the scope of mandatory. Legal Reference: Law No. 09-08 (CNDP): <https://www.cndp.ma/loi-09-08/> (accessed on 10 June 2026).

Informed Consent Statement: Informed consent was obtained from all participants involved in the study. Participation was voluntary, and participants were informed about the study title, researcher identity, study purpose, procedures, approximate duration, confidentiality measures, anonymized academic use of the data, right to withdraw at any time without penalty, and researcher contact information. By confirming the statement of consent, participants acknowledged that they had read the provided information and voluntarily agreed to participate in the study. This survey is administered online. The first page recalls the informed consent information, and the survey can only be started after the participant has read this page and checked the consent box below.

Data Availability Statement: The raw data supporting the conclusions of this article are included in the Supplementary Materials.

Acknowledgments: The author thanks the reviewers for their constructive feedback and insightful suggestions, which greatly improved the quality and clarity of this research.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

KPI	Key Performance Indicator
LLM	Large Language Model
QA	Question Answering
SOA	Service Oriented Architecture

References

1. Yusuf, H.; Money, A.; Daylamani-Zad, D. Pedagogical AI conversational agents in higher education: A conceptual framework and survey of the state of the art. *Educ. Technol. Res. Dev.* **2025**, *73*, 815–874. [[CrossRef](#)]
2. Akçapınar, G.; Sidan, E. AI chatbots in programming education: Guiding success or encouraging plagiarism. *Discov. Artif. Intell.* **2024**, *4*, 87. [[CrossRef](#)]
3. Schorcht, S.; Peters, F.; Kriegel, J. Communicative AI Agents in Mathematical Task Design: A Qualitative Study of GPT Network Acting as a Multi-professional Team. *Digit. Exp. Math. Educ.* **2025**, *11*, 77–113. [[CrossRef](#)]
4. Ghamati, K.; Banitalebi Dehkordi, M.; Zaraki, A. Towards AI-Powered Applications: The Development of a Personalised LLM for HRI and HCI. *Sensors* **2025**, *25*, 2024. [[CrossRef](#)] [[PubMed](#)]
5. Diebel, C.; Goutier, M.; Adam, M.; Benlian, A. When AI-Based Agents Are Proactive: Implications for Competence and System Satisfaction in Human–AI Collaboration. *Bus. Inf. Syst. Eng.* **2025**, *68*, 259–278. [[CrossRef](#)]
6. Gao, S.; Fang, A.; Huang, Y.; Giunchiglia, V.; Noori, A.; Schwarz, J.R.; Ektefaie, Y.; Kondic, J.; Zitnik, M. Empowering biomedical discovery with AI agents. *Cell* **2024**, *187*, 6125–6151. [[CrossRef](#)] [[PubMed](#)]
7. Zhao, X.; You, W.; Zheng, Z.; Shi, S.; Lu, Y.; Sun, L. How Do Consumers Trust and Accept AI Agents? An Extended Theoretical Framework and Empirical Evidence. *Behav. Sci.* **2025**, *15*, 337. [[CrossRef](#)] [[PubMed](#)]
8. Thapa, S.; Shiwakoti, S.; Shah, S.B.; Adhikari, S.; Veeramani, H.; Nasim, M.; Naseem, U. Large language models (LLM) in computational social science: prospects, current state, and challenges. *Soc. Netw. Anal. Min.* **2025**, *15*, 4. [[CrossRef](#)]
9. Catak, F.O.; Kuzlu, M. Uncertainty quantification in large language models through convex hull analysis. *Discov. Artif. Intell.* **2024**, *4*, 90. [[CrossRef](#)]

10. DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv* **2025**, arXiv:2501.12948. . [[CrossRef](#)]
11. Sarsur, D.; Alves, L.V.; Pena, P.N. Using Markov Decision Process over Local Modular Supervisors for Planning Problems. *IFAC-PapersOnLine* **2024**, *58*, 126–131. [[CrossRef](#)]
12. Bäuerle, N.; Jaśkiewicz, A. Markov decision processes with risk-sensitive criteria: An overview. *Math. Methods Oper. Res.* **2024**, *99*, 141–178. [[CrossRef](#)]
13. Jonsson, A. An axiomatic approach to Markov decision processes. *Math. Methods Oper. Res.* **2023**, *97*, 117–133. [[CrossRef](#)]
14. Vitetta, A. Sustainable Mobility as a Service: Framework and Transport System Models. *Information* **2022**, *13*, 346. [[CrossRef](#)]
15. Marchesi, L.; Goldman, A.; Lunesu, M.I.; Przybyłek, A.; Aguiar, A.; Morgan, L.; Wang, X.; Pinna, A. (Eds.) *Agile Processes in Software Engineering and Extreme Programming—Workshops: XP 2024 Workshops, Bozen-Bolzano, Italy, June 4–7, 2024, Revised Selected Papers*; Lecture Notes in Business Information Processing; Springer Nature: Cham, Switzerland, 2025; Volume 524. [[CrossRef](#)]
16. Shi, T.; Zhai, J.; Wang, H.; Chen, Q.; Zhai, M.; Hao, Z.; Yang, H.; Chen, W. GraphSet: High Performance Graph Mining through Equivalent Set Transformations. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, Denver, CO, USA, 12–17 November 2023*; ACM: New York, NY, USA, 2023; pp. 1–14. [[CrossRef](#)]
17. Schenker, A.; Bunke, H.; Last, M.; Kandel, A. A Graph-Based Framework for Web Document Mining. In *Document Analysis Systems VI*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3163, pp. 401–412. [[CrossRef](#)]
18. Khan, I.; Bokhari, M.U.; Afzal, S.; Alam, S. Distance driven graph neural network for advanced node classification through feature augmentation. *Discov. Comput.* **2025**, *28*, 70. [[CrossRef](#)]
19. Fernandes, D.; Matos-Carvalho, J.P.; Fernandes, C.M.; Fachada, N. DeepSeek-V3, GPT-4, Phi-4, and LLaMA-3.3 Generate Correct Code for LoRaWAN-Related Engineering Tasks. *Electronics* **2025**, *14*, 1428. [[CrossRef](#)]
20. Wu, X.K.; Chen, M.; Li, W.; Wang, R.; Lu, L.; Liu, J.; Hwang, K.; Hao, Y.; Pan, Y.; Meng, Q.; et al. LLM Fine-Tuning: Concepts, Opportunities, and Challenges. *Big Data Cogn. Comput.* **2025**, *9*, 87. [[CrossRef](#)]
21. Chen, Y.J.; Jhong, B.G.; Chen, M.Y. A Real-Time Path Planning Algorithm Based on the Markov Decision Process in a Dynamic Environment for Wheeled Mobile Robots. *Actuators* **2023**, *12*, 166. [[CrossRef](#)]
22. Bizzarri, F.; Mocenni, C.; Tiezzi, S. A Markov Decision Process with Awareness and Present Bias in Decision-Making. *Mathematics* **2023**, *11*, 2588. [[CrossRef](#)]
23. Yue, H.; Liu, J.; Zhang, Q. Applications of Markov Decision Process Model and Deep Learning in Quantitative Portfolio Management during the COVID-19 Pandemic. *Systems* **2022**, *10*, 146. [[CrossRef](#)]
24. Deng, L.; Zhong, Q.; Song, J.; Lei, H.; Li, W. LLM-Based Unknown Function Automated Modeling in Sensor-Driven Systems for Multi-Language Software Security Verification. *Sensors* **2025**, *25*, 2683. [[CrossRef](#)] [[PubMed](#)]
25. You, Y.; Liu, Z.; Wen, X.; Zhang, Y.; Ai, W. Large Language Models Meet Graph Neural Networks: A Perspective of Graph Mining. *Mathematics* **2025**, *13*, 1147. [[CrossRef](#)]
26. Paixao, L.D.A.G.; Rogers, W.P.; De Jesus, E.B. A Novel Methodology to Develop Mining Stope Stability Graphs on Imbalanced Datasets Using Probabilistic Approaches. *Mining* **2025**, *5*, 24. [[CrossRef](#)]
27. Zota, R.D.; Bărbulescu, C.; Constantinescu, R. A Practical Approach to Defining a Framework for Developing an Agentic AIOps System. *Electronics* **2025**, *14*, 1775. [[CrossRef](#)]
28. Gao, J.; Qiao, H.; Zhang, Y. Intelligent Recognition of Tool Wear with Artificial Intelligence Agent. *Coatings* **2024**, *14*, 827. [[CrossRef](#)]
29. Aylak, B.L. SustAI-SCM: Intelligent Supply Chain Process Automation with Agentic AI for Sustainability and Cost Efficiency. *Sustainability* **2025**, *17*, 2453. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.