AI

Systematic Review

Explicit and Implicit Learning Mechanisms in AI Educational Assistants: A Systematic Review

Fatmah Alqarni [1,*], Nada Alhirabi [2], Omer Rana [1] and Charith Perera [1]

[1] Computer Science and Informatics, Cardiff University, Cardiff CF24 4AG, UK; ranaof@cardiff.ac.uk (O.R.); pererac@cardiff.ac.uk (C.P.)
[2] Department of Computer Science and Engineering, College of Applied Studies, King Saud University, Riyadh 11451, Saudi Arabia; nalhirabi@ksu.edu.sa
* Correspondence: alqarnif@cardiff.ac.uk

Abstract

Artificial intelligence techniques have made notable progress in supporting learning processes, with increasing adoption across educational contexts. However, despite the increasing work on AI-assisted techniques, explicit and implicit learning mechanisms in AI educational assistants have not been systematically categorised. The study of how these techniques aid in and are implemented for learning remains underexplored. Therefore, a more systematic categorisation of how these techniques support learning through user interaction is needed. This paper presents a systematic review of 38 studies published between 2000 and 2024, spanning domains including programming education, cognitive skills, language learning, and the AI field. This review was conducted and reported in accordance with the PRISMA 2020 guidelines. In this review, we propose a taxonomy of explicit and implicit learning features. We analyse implementation aspects (e.g., knowledge representation, algorithms, and interaction modalities) and synthesise how prior work evaluates learning support capabilities. The findings show that (i) 79% of reviewed studies support explicit and 21% supported implicit learning through interaction; (ii) written interaction dominates (45%), followed by visualisation (34%), while voice-based interaction remains underrepresented (9%); (iii) some implementations lack details (e.g., knowledge bases and validation methods); and (iv) evaluation practices remain uneven, with most studies relying on experiment evaluation, highlighting the need for robust evaluation practices.

Keywords: artificial intelligence (AI); LLM; generative AI; explicit learning; implicit learning

Academic Editors: Miguel Angel Cazorla and Andrej Flogie

Received: 26 February 2026
Revised: 6 April 2026
Accepted: 16 April 2026
Published: 1 May 2026

1. Introduction

Learning is a fundamental process for acquiring knowledge, developing skills, and supporting informed decision-making. As described by Sequeira [1], learning involves developing new abilities, reshaping perspectives, and understanding conceptual relationships. In recent years, artificial intelligence (AI) has increasingly supported learning through interactive systems, virtual environments, and conversational interfaces [2–5]. AI-based systems are often embedded in various educational contexts, with studies reporting increased learner engagement and improved learning outcomes [6].

AI techniques have evolved from simple rule-based instructional systems to more sophisticated approaches. These include adaptive intelligent tutoring systems, recommendation engines, conversational agents, and large language model (LLM)-based assistants. These systems differ not only in their technical implementation but also in how they

 https://doi.org/10.3390/ai7050160

support learning through user interaction. Some techniques provide direct instruction, structured feedback, or guided questioning, while others facilitate learning indirectly through questioning, experimentation, and interaction. However, despite the rapid development of AI-supported educational systems, explicit and implicit learning mechanisms in AI techniques have not been systematically categorised.

Learning supported by AI techniques can occur in two main forms: explicit and implicit learning. Explicit learning involves consciously engaging with explanations, feedback, and structured instruction to acquire knowledge. Implicit learning, in contrast, refers to knowledge acquired unintentionally through activities, interaction, and experiential engagement. While both forms have been discussed in educational theory, their manifestations in AI educational techniques have not been systematically categorised. The existing literature often evaluates systems within specific domains or focuses primarily on technical implementation, leaving a gap in understanding how interaction design patterns function as learning mechanisms.

This systematic review addresses this gap by synthesising how AI educational assistants support learning through interaction-driven mechanisms. We classify explicit and implicit learning mechanisms, analyse how these mechanisms are implemented (e.g., through conversational AI, tutoring systems, recommendation systems, and AI assistants), and examine how prior studies evaluate their effectiveness. By integrating learning theory with implementation analysis, this review provides a structured foundation for understanding how AI systems transform educational processes.

The contributions of this paper are as follows:

- A taxonomy of explicit and implicit learning mechanisms in AI educational assistants;
- Systematic mapping between learning mechanisms, implementation approaches, and AI techniques;
- A synthesis of evaluation practices and research gaps to inform the design and assessment of AI-enhanced educational systems.

The remainder of this paper is organised as follows: Section 2 presents the methodology used to select and analyse the literature. Section 3 presents a motivating example to illustrate the need for AI-supported interaction mechanisms in complex learning contexts. Section 4 synthesises explicit and implicit learning mechanisms in AI education techniques. Section 5 discusses how these techniques are implemented. Section 6 analyses the underlying AI techniques, covering knowledge representation, algorithms, evaluation, interaction types, and the role of large language models and generative AI in education. Section 7 discusses the limitations and research directions. Finally, Section 8 concludes the literature review.

2. Methodology

This review follows the PRISMA guidelines [7] and the snowballing procedure outlined by Kitchenham and Brereton [8] and Wohlin [9]. The review protocol was retrospectively registered in the Open Science Framework (OSF) (registration ID: https://osf.io/8x4ed/ (accessed on 26 March 2026)). The PRISMA checklist used to guide the review process is provided as a Supplementary Material.

2.1. Search Strategy and Information Sources

The screening and selection process was conducted by the first author, and the results were then reviewed and verified by the fourth author at each stage. The first step was to identify keywords and phrases to formulate search strings; Boolean operators, such as "AND" and "OR", were used to combine strings and acquire more relevant papers (see Figure 1). We conducted the initial search across all selected databases between June and

September 2022, followed by an update in November 2023. To capture recent advances in LLM-based educational assistants, we conducted an additional search between September and December 2024. Full search strategies for each database and time period are available in the Supplementary Materials. Subsequently, we searched for relevant publications in the following databases: Google Scholar, ACM, IEEE, MDPI, Springer, and Taylor & Francis. The search was conducted using three methods: automatic, manual, and snowball. Google Scholar was used initially as a starting point to avoid bias in favor of any specific publisher. To further refine the search, we conducted a manual search using the keywords mentioned in Figure 1. When the paper was relevant, we conducted forward and backward snowballing to find more relevant articles, as seen in Figure 2.

The initial search identified a total of 16,335 records across all databases: Google Scholar (automatic search, n = 15,900) and manual searches including ACM Digital Library (n = 180), IEEE Xplore (n = 120), MDPI (n = 45), Springer (n = 60), and Taylor & Francis (n = 30). Before screening, 5685 duplicate records were removed and a further 2500 records were removed by applying date filters. This resulted in 8150 records for title screening.

After title screening, 8030 records were excluded, leaving 120 studies. These were combined with 11 records identified through backward and forward snowballing, resulting in 131 reports sought for retrieval. All reports were successfully retrieved. The reports were then selected for abstract and conclusion screening, leading to the exclusion of 73 articles, resulting in 58 studies assessed for full-text eligibility. Of these, 20 were excluded based on inclusion and exclusion criteria, resulting in 38 studies included in the final review.

A formal study-level risk of bias assessment was not conducted. However, to reduce bias in the review process, several measures were adopted. First, the screening and selection of papers were reviewed and discussed by the second and fourth authors to ensure inter-reviewer reliability. Second, the initial search was performed using Google Scholar rather than a single discipline-specific database, thereby reducing the risk of publication bias toward any particular journal or publisher. Therefore, the results of this review should be viewed as general qualitative trends, rather than providing precise quantitative measures of impact.

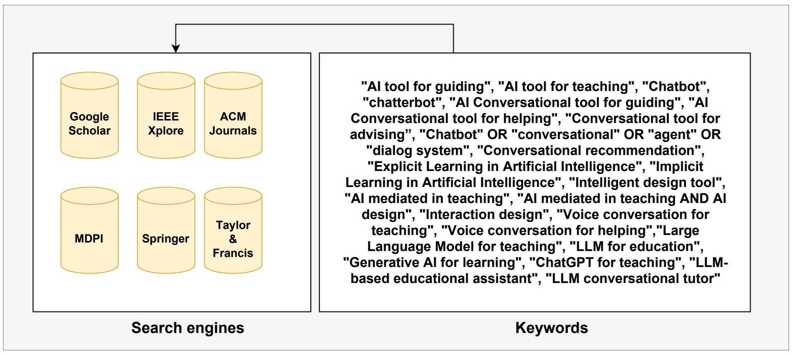


Figure 1. Search engines used a combination of keywords and terms to explore relevant existing studies on intelligent design techniques.

2.2. Inclusion and Exclusion Criteria

In each phase of the search process after the initial search, reading the title, abstract, and conclusion of a paper was necessary to make a decision to either exclude or extract the paper. In case there were duplicate papers, the most extensive or recent version was considered. The criteria for exclusion were the study (i) was not written in English; (ii) does not mention AI, learning, privacy, or design or addresses these concepts in general terms; (iii) did not provide learning techniques or help for users through AI; and (iv) lacked a full version (such as a poster or abstract only). Finally, since the scope of AI is wide, robotics was excluded as it falls outside the boundaries of this particular search.

If a paper was not excluded, then it was evaluated by inclusion criteria. We discarded the paper if none of the inclusion criteria were met. The inclusion criteria were the study (i) discusses learning with AI techniques and (ii) discusses how the AI techniques are implemented. All relevant papers reviewed were published between 2000 and 2024.

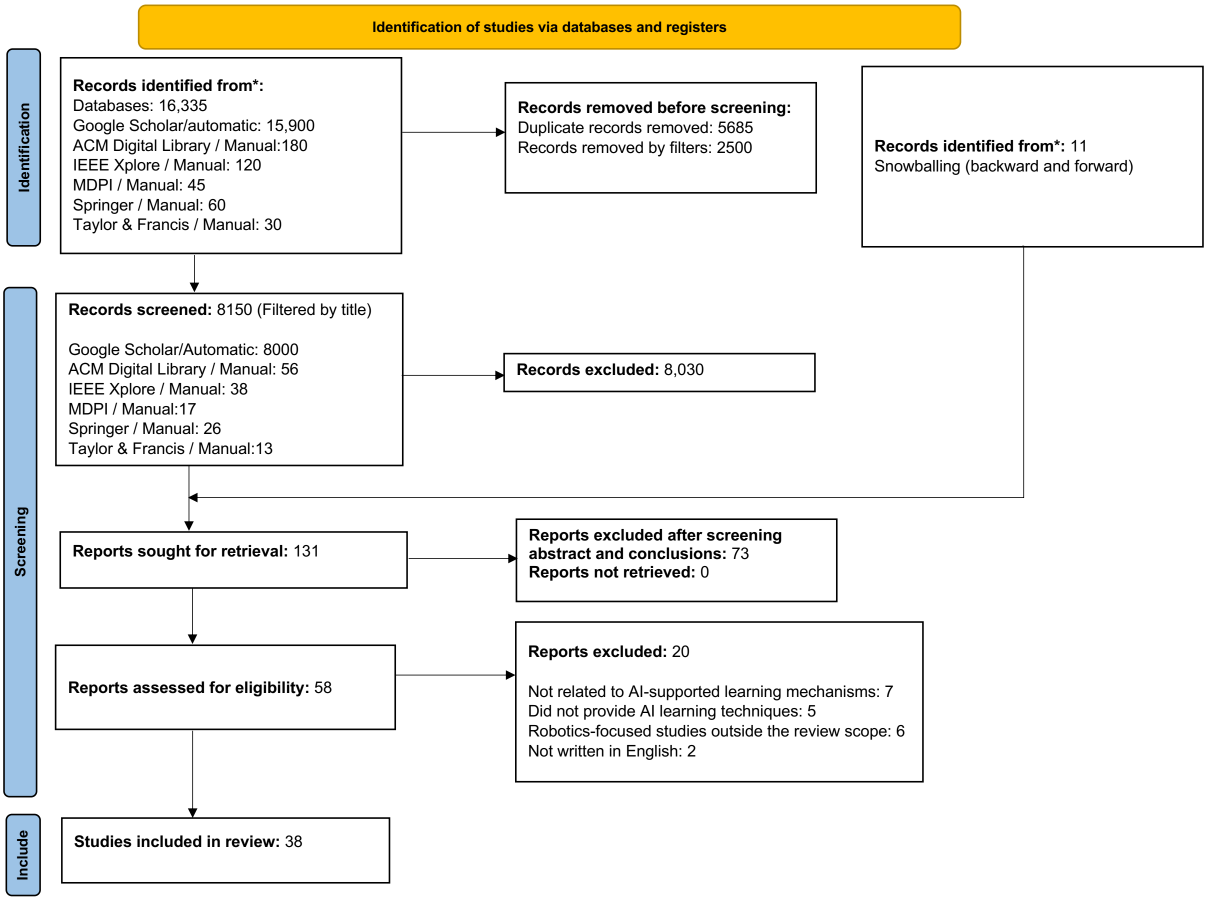


Figure 2. PRISMA 2020 flow diagram of the study selection process, (*) indicates records identified from databases and through snowballing (backward and forward).

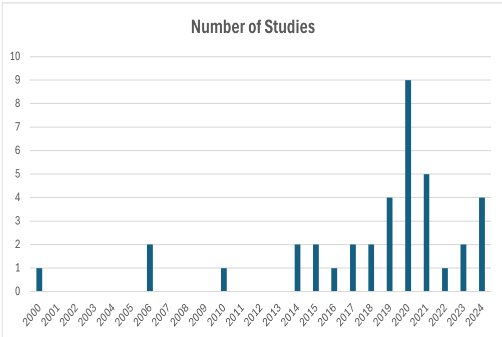
2.3. Data Extraction

Data extraction was conducted systematically using a predefined extraction form developed by the first author and reviewed by the fourth author, following established guidelines for systematic reviews [8,10]. Prior to full application, the extraction form was reviewed and discussed between the first and fourth authors through an iterative process of document exchange, ensuring clarity and consistency across all extracted variables.

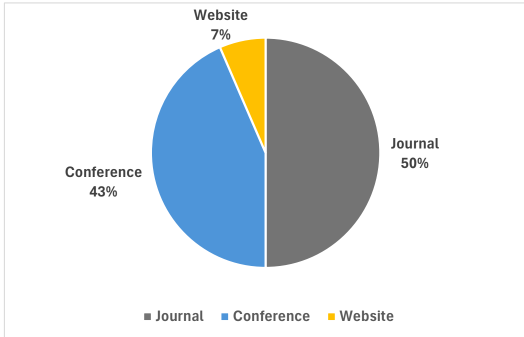
Of the 38 included studies, 34 were subject to full data extraction across all variables. The remaining four studies were referenced exclusively in the motivating example (Section 3) and were therefore not included in the full extraction process. The following variables were extracted from each included study: (1) publication year; (2) venue type (e.g., conference, journal, or website); (3) publisher; (4) domain; (5) type of learning supported (explicit or implicit); (6) interaction features and mechanisms; (7) implementation approach (e.g., chatbot, tutoring system, or AI tool); (8) knowledge representation and update strategy; (9) target user group interacting with the system; (10) algorithms and techniques used; (11) interaction modality (written, spoken, visual, or action-based); (12) evaluation approach, including evaluation type (quantitative, qualitative, or mixed);

(13) data collection methods; (14) number of participants; and (15) metrics. The complete extraction form and extracted data are provided as Supplementary Materials.

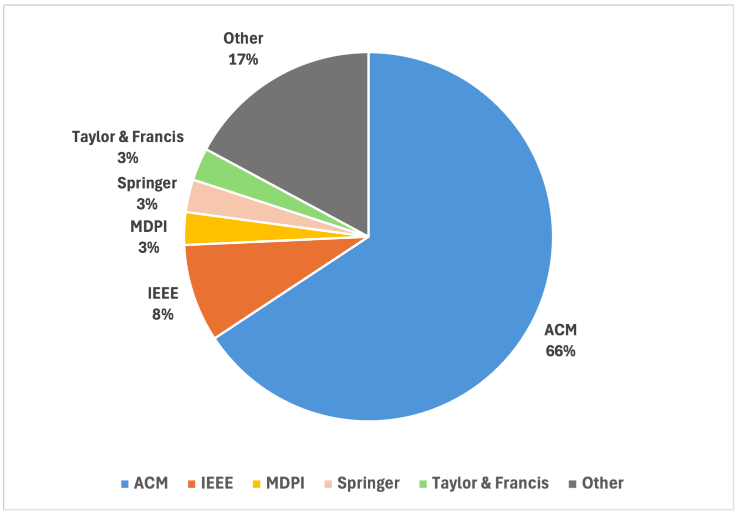
In this survey, we analysed a set of papers from various publications. Figure 3 provides a detailed summary of the reviewed studies. Figure 3a illustrates the chronological distribution of publications from 2000 to 2024. The review of the extant literature indicates that the number of studies increased gradually from the early 2000s through the 2010s. This trend culminated in a marked peak between 2019 and 2020. Although there was only a slight decrease in the following years, the numbers remained higher than in the early stages, indicating sustained interest in the domain. Figure 3b reveals that 43% of the papers originated from conference proceedings, while journal articles accounted for 50% and website sources comprised 7%. Finally, Figure 3c presents the percentage of studies per publication venue, with most studies selected from high-quality sources.



(**a**)



(**b**)



(**c**)

Figure 3. An extensive review of high-quality studies over the period from 2000 to 2024. Number of reviewed studies per year (2000–2024) (**a**); distribution of publication types (conference, journal, and website) (**b**); the proportion of chosen studies by publication venue (**c**).

2.4. The Aim of the Systematic Literature Review

The main purpose of this review is to answer the following research questions (RQs):

- RQ1: What interaction mechanisms in AI educational assistants support explicit and implicit learning? This research question discusses different learning mechanisms that end users acquire through explicit or implicit AI techniques. Learning can occur explicitly, where users are aware, or implicitly, without the users realising it. The details are explained in Section 4.

- RQ2: How have AI assistant techniques been implemented in the literature? (i) How is knowledge stored and continuously updated? (ii) Who are the stakeholders interacting with the system? (iii) What algorithms and techniques are used? (iv) How do they evaluate? (v) What types of output are generated?
  This research question addresses the methods developers use to implement AI techniques, including how they store and update knowledge, how users interact with these techniques, the algorithms and methods employed, the system evaluation process, and finally the types of output generated. Sections 5 and 6 include more details.

3. A Motivating Example

Some domains of study are inherently complex and require structured suggestions, particularly for novice learners who need structured support. Domains such as privacy by design and security demand not only conceptual understanding but also the ability to apply privacy during the design of internet of things (IoT) systems [11]. However, many existing tools in these domains provide static workflows or rule-based feedback, often assuming substantial prior knowledge, as seen in [12,13]. As a result, novice learners may struggle to understand underlying concepts or to integrate them meaningfully into their design decisions. To make this concrete, consider a novice developer designing a smart home application that integrates IoT components. While designing the application, the developer must consider issues such as data minimisation, user consent, and encryption. Without adequate structured guidance, these privacy issues may remain shallow or be ignored completely. An AI-supported educational assistant could provide explicit features (e.g., guided questioning, feedback, or warnings), as well as implicit features (e.g., learning through interaction and iterative refinement). This supports learners in developing a deeper understanding during the design process.

This example demonstrates the broader need for AI-supported interaction features to guide learners through complex, cognitively demanding domains. The following Section 4 presents how AI educational assistants implement explicit and implicit learning features across diverse contexts and analyse the technical approaches used to realise these capabilities.

4. Explicit and Implicit Learning in AI Techniques

As discussed in Section 1, learning underpins the acquisition of knowledge and the achievement of goals across domains. Benton [14] mentioned that human cognition extends beyond internally stored knowledge and relies on external sources, including the environment and interaction with others. Kohda [15] further suggests that human performance can be enhanced when supported by AI systems. Within this interactional framework, learning through AI may occur explicitly through direct instruction, explanation, and feedback. It could also be implicit by performing activities and receiving feedback (see Figure 4).

The taxonomy presented in Sections 4.1 and 4.2 was developed using inductive analysis of the reviewed literature. Explicit and implicit learning were first defined based on the established learning science literature [16–19]. These definitions served as the theoretical anchor for classification. The learning features identified in each study were then examined based on their functional role, specifically whether users consciously engaged with the AI technique (explicit) or whether learning was described as occurring incidentally through interaction without deliberate awareness (implicit). This criterion was applied consistently across all reviewed studies as the basis for classification. Features were subsequently grouped by conceptual similarity within each category. Where the same surface interaction type (e.g., feedback and questioning) appeared in both explicit and implicit categories, this reflects its differing functional role across study contexts, not an inconsistency in classification.

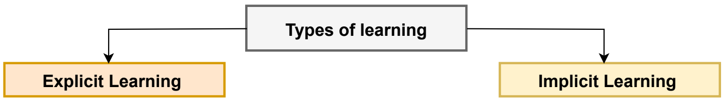


Figure 4. Types of learning in AI techniques.

4.1. Explicit Learning

Explicit learning is defined as a type of learning in which learners are directly instructed or taught for a particular purpose [16]. It focuses on the verbal explanation of principles, rules, and concepts and is often used to gain a better understanding or knowledge of a specific topic. This type of learning enables users to realise their aims and objectives to achieve their goals [16]. As such, Ellis [17] mentioned that it requires conscious effort and results in knowledge that is represented in explicit form. Moreover, it provides users with the necessary guidance and structure to help them progress towards their desired outcome. The following sections provide an in-depth exploration of the different features of explicit learning that perform between humans and AI and the ways in which users may become involved in it.

4.1.1. Asking Questions

Questions represent one of the most direct features through which AI educational techniques support explicit learning. In programming contexts, this has taken the form of intelligent assistants that answer learner queries about core concepts such as data structures and programming definitions [20]. Reported increases in student motivation and enjoyment were noted as a result. This question-asking feature has also been developed for broader educational settings, where virtual teaching assistants generate tailored responses derived from instructor materials [21]. This simultaneously enables instructors to identify individual student needs [21]. A more structured approach is evident in systems that combine question-asking with immediate performance-based feedback, directing learners toward more challenging tasks or corrective content depending on their responses [22]. Across these studies, questioning functions not only as a means of information retrieval but also as a feature for actively guiding learner progression.

4.1.2. Yes or No Questions

Yes/No questions represent a simplified yet effective form of explicit interaction, particularly suited to early learners. In one of the AI assistance systems designed for preschool children, a dialogue-based system supported the identification, replication, and writing of integer numbers from zero to nine [23]. Interactions were structured around voice-based yes/no responses, reducing cognitive load for young users. Improved motor skills in writing were observed among children who engaged with the system, indicating that even simple binary questioning can yield measurable learning outcomes.

4.1.3. Multiple Choice Questions

Multiple-choice questions support explicit learning by presenting learners with structured options, guiding them to consider and select the most appropriate answer. This approach has been applied in cognitive and educational domains. In circuit design, one system generates multiple circuit options on screen, allowing novices to select based on criteria such as cost and component availability [24]. Users of this interactive system completed tasks more successfully and efficiently than those without access to it. A similar feature has been adopted in language learning, where learners select from multiple-choice options to answer questions [25]. Users are noting measurable improvement in their English language skills, as reported in the study.

4.1.4. Answering Questions

Answering questions is another explicit learning feature that engages learners actively with educational content. In language learning contexts, this has been implemented through tools that provide quizzes, vocabulary and grammar lessons [26]. Learners are required to respond to structured questions as part of the learning process. This approach has also been extended to higher education settings, where both technical and non-technical educators engage with course content through a question and answer feature [27].

However, across the reviewed studies, question and answer interactions are often limited to pre-set answers. This may make more in-depth exploration harder and make the AI techniques less adaptive in different and complex learning situations.

4.1.5. Guidance

Guidance represents a type of explicit learning support, particularly valuable for novice learners navigating complex technical domains. Step-by-step instructional guidance has been shown to support learners in building circuits and writing code [28]. Through this guidance, learners receive instructions progressively, reducing the likelihood of errors. Increased confidence in completing technical tasks was reported among users who engaged with this form of guided interaction [28]. From the authors' perspective, step-by-step instructions may reduce autonomous reasoning skills, as learners may become overly reliant on structured guidance.

4.1.6. Warning Message

Warning messages represent an explicit learning feature that alerts learners to errors in real time, enabling them to identify and correct mistakes promptly. This feature has been applied in programming education, where interactive environments provide warning messages when errors occur during tasks [29]. Beginner programmers engaging with this feature reported improvement in their data structure skills and knowledge [29]. However, we argue that warning messages alone may be insufficient for deep learning, as alerting learners to errors without explaining the underlying reasoning may limit conceptual understanding.

4.1.7. Generate Tasks or Solve Tasks

Generating and solving tasks represents as explicit learning feature, engaging learners actively in domain-specific activities. Across the analysed studies, this feature has been found in several distinct contexts. In programming and circuit design, task-based learning has taken the form of debugging and assembly exercises, where learners follow predetermined steps to write code or build circuits [30]. This structured approach supports learners in understanding content while reducing errors. In education, animated visualisation tools have been used to present concepts such as wireless network attacks [31]. Learners engage by controlling the animation process and receiving dynamic descriptions, with positive feedback reported regarding ease of use and conceptual understanding. Task generation has also been applied in reading comprehension, where knowledge graphs are used to automatically generate test questions associated with text [32]. A similar activity-based approach is evident in education, where learners engage with quizzes and image analysis tasks to develop knowledge of complex clinical concepts [33]. Learners have been supported through tools that allow them to modify parameters and develop complex visualisations [34,35]. Positive motivation and improved understanding of computer graphics concepts were reported among users. Finally, in experimental design contexts, tools have enabled learners to create playable scenarios by modifying text and symbols [36], though the need for AI assistance to support successful task completion was noted.

https://doi.org/10.3390/ai7050160

4.1.8. Suggestions

Suggestions represent an explicit learning feature through which AI systems proactively offer hints, recommendations, or relevant materials to guide learners. Suggestion-based support has been implemented through chatbot interfaces that provide hints and conceptual explanations to learners [26]. Favourable feedback was reported, though limitations were noted when users attempted voice-based interaction. Beyond language learning, suggestion features have been applied in creative and design contexts. AI-based tools have been used to help designers quickly identify relevant ideas and visual inspiration during the design phase [37]. Collaborative design settings have further extended this approach, where AI suggestions enable remote designers to share and build upon each other's thoughts [38]. Overall, the suggestion feature accelerates the creative process by reducing the time needed to find relevant inspiration across these studies.

4.1.9. Alerting

Alerting represents an explicit learning feature through which AI systems notify learners of potential issues or upcoming events. This has been implemented in educational settings to inform students about upcoming deadlines, ensuring timely engagement with course requirements [21].

4.1.10. Feedback

Feedback is a well-established mechanism through which learners determine their level of success and identify areas for improvement [39]. Digital tools have been shown to deliver feedback more efficiently than traditional pen-and-paper methods [40]. Across the reviewed studies, feedback has been implemented in several distinct forms. In programming education, feedback has been delivered through conversational assistants that respond to learner queries about coding exercises and provide assessment on submitted assignments [41]. A more structured approach is evident in systems that offer multiple feedback types simultaneously, including syntax error feedback, execution feedback, and both reactive and proactive guidance [42]. In literacy education, feedback has been used to support deaf learners by automatically identifying grammatical errors in written text and presenting them for correction [43]. This approach enables learners to resubmit their work iteratively, reducing the social pressure associated with making mistakes in front of a human teacher. In engineering education, correctness-based feedback has been applied to problem-solving tasks, informing learners whether their steps are accurate without revealing the answer [44]. In academic advising contexts, feedback has been used to guide students in selecting institutions and majors aligned with their interests [45]. Performance-based feedback has also been integrated with question-asking mechanisms, where learner responses are assessed and used to direct them toward more challenging or corrective content [22]. Finally, in design education, feedback has supported the transformation of low-fidelity sketches into higher-quality outputs, streamlining the creative workflow [46]. Across all these contexts, feedback plays a role not as a final judgement but as a dynamic tool for guiding ongoing learning.

4.1.11. Editing Feedback

Editing feedback represents an explicit learning feature that enables learners to receive feedback and make necessary modifications based on it. This approach has been implemented in education, where interactive environments support novice programmers in learning data structures. One system provides a visually intuitive representation of data structures, allowing learners to quickly grasp their components [29]. Another enables learners to create diagrams of data structures and have code automatically generated from

https://doi.org/10.3390/ai7050160

them [47]. Together, these tools demonstrate how editing feedback can bridge visual understanding and practical coding skills. Figure 5 shows an overview of the various forms of explicit learning in AI techniques. These features can assist in engaging a user in active learning and help them understand the knowledge. However, we argue that although editing feedback helps in iterative learning, the reviewed implementations are limited to programming contexts. Evidence of its effectiveness in other domains remains absent from the literature.

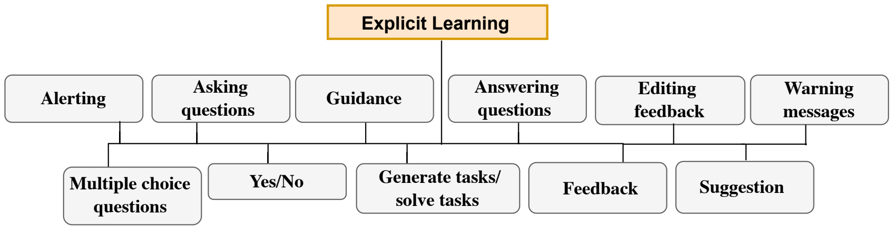


Figure 5. A taxonomy derived from user interactions with AI techniques to support explicit learning.

4.2. Implicit Learning

Implicit learning is a second type of learning that is completely different from explicit learning. It is un-deliberate learning, and learners' engagement is unintentional [16]. This form of learning occurs incidentally through engagement with activities and interaction, rather than through conscious instruction [18]. The concept was first explored by [48], whose work demonstrated that learners can develop an understanding of structural relations without using strategies or being able to express the knowledge in words. Building on this foundation, implicit learning has been further characterised as a process in which learners are unaware that they are acquiring knowledge through the activities they undertake [19].

4.2.1. Asking and Answering Questions

Asking and answering questions can support implicit learning when learners engage in dialogue without a deliberate intention to learn. In e-learning contexts, voice-based assistants have been used to facilitate skill development through conversational interaction. One such implementation adopts a flexible role, functioning either as a peer or an adviser depending on the learner's responses [49]. Thereby naturally guiding the conversation toward skill improvement [49]. This approach has also been extended to collaborative problem-solving, where groups of learners engage with a voice assistant to think through and discuss problems [50]. Compared to groups interacting with human tutors, those using voice assistance demonstrated more homogeneous contributions and showed evidence of acquiring collaboration and problem-solving skills. Taken together, the act of asking and answering questions facilitated learning incidentally, without learners necessarily being aware of the knowledge they were acquiring.

4.2.2. Performing Activities

Performing activities represents a form of implicit learning in which learners acquire knowledge through engagement with tasks, without a deliberate intention to learn. Activity-based learning has enabled students to solve exercises using visual programming tools, fostering an understanding of computational thinking and algorithm behaviour [51]. A similar approach has been adopted to introduce machine learning concepts to beginner learners without requiring prior coding knowledge [52]. Through hands-on interaction, learners develop an intuitive understanding of classification processes in an accessible way. Activity-based implicit learning has also been extended to younger audiences. Conversational agents have been designed to allow children to teach and train an AI system with facts about animals and then observe how the system responds to questions [53].

This playful interaction supports understanding of machine learning and knowledge representation without explicit instruction. Similarly, platform-based tools have enabled non-technical users to add datasets, train models, and evaluate outputs [54,55]. Taken together, these activity-based implementations encourage learners to think critically about AI processes through experimentation rather than direct instruction. However, it is worth noting that performing activities as an implicit learning feature is predominantly applied in AI education contexts, leaving its potential in other domains largely unexplored.

4.2.3. Feedback

Implicit feedback supports learning by providing learners with immediate responses to their actions, enabling them to adjust their approach without being explicitly instructed to do so. Conversational AI has been used to offer real-time feedback on pronunciation accuracy, allowing learners to improve their speaking ability through natural interaction [56]. This approach supports skill development in a shorter time than traditional methods. Implicit feedback has been also applied in creative design contexts. Designers evaluate and refine their decisions by observing AI-generated outputs that reflect their design choices [57]. A related approach enables designers to experiment with multimodal inputs such as images, colours, and text, learning implicitly by observing the impact of these inputs on generated outputs [58]. These implementations demonstrate that implicit feedback supports learning across diverse domains by encouraging experimentation and reflection rather than direct instruction. However, implicit feedback is inherently dependent on learner engagement, a factor that is insufficiently addressed in the reviewed studies.

Figure 6 illustrates the different types of implicit learning that can be employed to educate learners without any deliberate intent to learn. Table 1 provides a glimpse into the explicit and implicit learning techniques used in AI assistants and the domain to which they belong. As presented in Table 1, 27 studies (79%) were classified as supporting explicit learning features and 7 studies (21 %) as supporting implicit learning features. Importantly, while each study was categorised according to a dominant learning type, several interaction features (e.g., question-answering and feedback) were observed across both explicit and implicit contexts, indicating overlap at the learning features rather than at the study level.

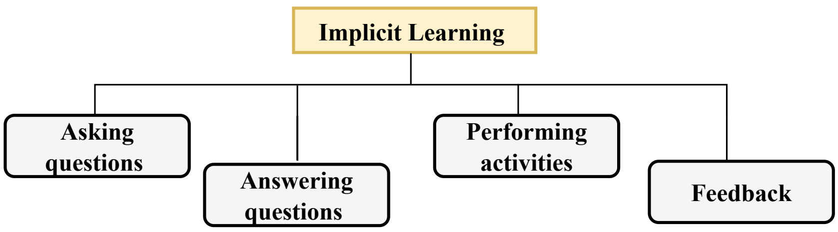


Figure 6. A taxonomy derived from user interactions with AI techniques to support implicit learning.

In addition, Table 1 reveals a discernible temporal pattern across the 2000–2024 period. The earliest studies [20,31,34,35,42–44,47] are exclusively associated with explicit learning, particularly feedback, and structured question-answering, with domains concentrated in programming education and literacy. A transition toward implicit learning became evident during 2019–2020, notably through voice-based systems such as Alexa for e-learning [49,50,52,53], while explicit learning continued to dominate. The most recent period (2021–2024) is marked by the integration of LLMs into systems such as [22,27,57,58], where the boundary between explicit and implicit learning becomes less distinct, as systems increasingly support both features simultaneously. Overall, implicit approaches became more prevalent after 2019, reflecting a broader shift from structured instruction toward more flexible, interaction-driven learning support.

Table 1. Summarised explicit and implicit learning in AI techniques across various domains, presented in a greyscale to bluescale spectrum: commencing with **Programming** and progressing to **Education**, **Cognitive**, **Skills**, **AI**, and **Computer Science** (in bluescale). A checkmark (✓) indicates that the study supports the corresponding learning type.

Ref.	Year	Learning Type		Domain	Brief Description of Learning Types
		Explicit	Implicit		
Konecki et al. [20]	2015	✓		Programming	Asking questions/answering questions
Shekhar et al. [21]	2020	✓		Education	Allowing students to ask questions/Guiding students by providing answers/alerting about deadlines
Hobert [41]	2019	✓		Education	Providing feedback/providing suggestions to students
Villegas-Ch et al. [23]	2022	✓		Education	Answering yes/no questions
Atilola et al. [44]	2014	✓		Education	Providing feedback to engineering students
Michaud et al. [43]	2000	✓		Education	Providing feedback to users/allowing users to edit the feedback
Sun et al. [25], Allen et al. [27]	2021, 2024	✓		Education	Asking questions/answering Questions
Yuan et al. [31]	2010	✓		Education	Controlling the animation process
Peternier et al. [34], Dias et al. [35]	2006	✓		Education	Modifying parameters/developing tasks
Aldeman et al. [33]	2021	✓		Education	Performing activities
Pham et al. [26]	2018	✓		Education	Answering questions/suggesting hints
Imtiaz et al. [29]	2018	✓		Education	Providing feedback/performing activities/issuing warning messages
Buchanan and Laviola Jr [47], Fossati et al. [42]	2014, 2015	✓		Education	Providing feedback
Elragal et al. [22]	2024	✓		Education	Asking questions/providing feedback
Anderson et al. [24]	2017	✓		Cognitive	Answering questions
Drew et al. [30]	2016	✓		Cognitive	Performing tasks
Zhao et al. [32]	2021	✓		Cognitive	Generating tasks/solving tasks
Kim et al. [28]	2020	✓		Cognitive	Advising users/providing feedback/alerting users
Zhao et al. [49], Winkler et al. [50]	2020, 2019		✓	Skills	Asking questions/answering questions
Harteveld et al. [36]	2017	✓		Skills	Performing activities
Carney et al. [52], Lin et al. [53]	2020		✓	AI	Performing activities
Rodríguez-García et al. [55]	2021		✓	AI	Performing activities
Estevez et al. [51], García et al. [54]	2019, 2020		✓	AI	Performing activities
Koch et al. [37], Koch et al. [38]	2019, 2020	✓		AI	Providing suggestions
Chen et al. [57], Peng et al. [58]	2024		✓	AI and design	Providing feedback
Sermuga Pandian et al. [46]	2020	✓		Computer Science	Providing feedback

5. Implementation Approaches

To answer RQ2, this section discusses the implementation approaches used to develop the technique. To gain further insight into the implementation process, we analyse the implementation of papers mentioned in Section 4. Four main types of approaches are typically implemented, each with a specific purpose and designed to interact with users in an intelligent way, as illustrated in Figure 7. The next sections, Sections 5.1–5.4, discuss each of these approaches in details.

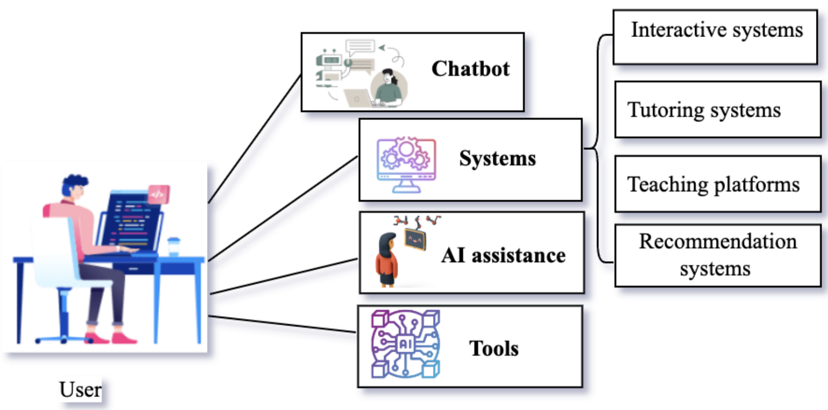


Figure 7. Types of implementation approaches and user interactions with AI techniques.

5.1. Conversational AI

Conversational AI has grown significantly over the last ten years, which aids humans in interacting virtually with computers. Conversational AI is known as a "conversational agent" in [53]; "task-oriented bots"; "task bots", in short [59]; and in some research, chatbots" [41,60]. A chatbot is defined as a user interface that is created to mimic chats with users online, typically through text or voice interactions [61]. Chatbots are designed to automate conversations and provide more efficient, accurate, and personalised services to users.

Rule-Based and AI Chatbots

Rule-based chatbots are designed with predefined rules, whereas AI chatbots can understand the context by learning from information gathered; it can be called a "conversational agent" or "machine learning chatbot" [60], which relies on AI, natural language processing, and machine learning. Although rule-based chatbots are inflexible, the reliability of these rules is one of the benefits of these rules because developers can create and delete rules for the purpose of editing or fixing errors. Training systems based on rules with extensive keywords has been recommended to ensure accurate outputs [62]. However, AI chatbots are more flexible and less reliable. The reason behind this is that AI chatbots can learn from the mistakes and answers of users in order to reply accurately, but the tolerance for grammatical mistakes is higher. Table 2 summarises the differences between rule-based chatbots and AI chatbots.

One approach combined a "chatbot-based system" built with natural language with an "intelligent programming tutor" to provide proper learning support [41]. This technique supports novice programmers in answering their questions and guiding them through programming tasks. Another implementation deployed a chatbot on mobile devices, enabling learners to interact through a chat window [26].

Table 2. Rule-based chatbots vs. AI chatbots.

Rule-Based Chatbots	AI Chatbots
Predefined rules	Learn from information gathered
Reliable	Less reliable
Less flexible	More flexible
Used in simple scenarios	Used in more complex scenarios

A contextual virtual teaching assistant was implemented using RASA, an open-source conversational AI platform [21]. This AI chatbot communicates with students and provides contextual support throughout their learning process.

5.2. AI Systems

Intelligent systems have seamlessly woven themselves into our daily lives, becoming an integral part of our routines [63]. These systems provide learning experiences by leveraging their capacity to assist users in acquiring knowledge. They encompass a wide range of functionalities, including interactivity, tutoring, teaching, and recommendation systems.

5.2.1. An Interactive System

An interactive system refers to a computer system in which users can interact with a running program by providing data or instructions using input devices such as a keyboard or mouse [64]. In electronics education, one such system was designed to help users with no prior knowledge design and build circuits [24]. A similar approach has been adopted in creative design contexts, where an interactive system allows designers to enter textual inputs, choose recommended design features, and manage the creative process through image-to-image and text to image comparisons [57].

5.2.2. Tutoring System

Intelligent tutoring systems are computerised educational platforms that offer individualised guidance by simulating and analysing the learning progress and behaviour of each learner [65–67]. Tutoring systems have been used to aid learners with written language skills [43]. In programming education, intelligent tutoring systems have been developed to provide automated feedback for learning data analysis and teaching linked lists [42]. A related implementation focused on teaching data structures more broadly, offering structured guidance to support learners in complex programming concepts [68].

5.2.3. Teaching Platform

AI offers a distinctive opportunity to tailor learning experiences, boost student engagement, and strengthen educators' capabilities by utilising machine learning algorithms and large-scale data [69]. In medical education, machine learning has been applied within an intelligent platform to support the teaching of complex clinical concepts [33]. A similar platform-based approach has been adopted in language education, where artificial intelligence and knowledge recommendation are combined to create a modern teaching tool for students [25].

5.2.4. Recommendation System

A recommender system is a combination of software tools and machine learning methods designed to offer valuable suggestions based on a user's preferences [70]. In reading comprehension, an intelligent system has been implemented to apply cognitive intelligence to teaching [32]. A related approach has been adopted in academic advising, where a recommender system identifies students' interests and skills to provide personalised guidance [45].

5.3. AI Assistance

AI assistance systems have been developed across a range of educational and creative contexts. In programming education, an intelligent assistant has been proposed to support students in learning programming languages [20]. In design contexts, AI-based assistance has been used to automate the transfer of low-fidelity sketches to higher-fidelity representations [46]. Cooperative platforms have also been developed to help designers quickly find inspiring images for their projects [37,38]. Beyond these contexts, AI assistance has been extended to support children in learning numbers [23].

5.4. AI Tools

Tools refer to a wide range of resources and methods that are essential for the practice of interaction design [71]. In the context of design practice, tools include items such as sketches, brainstorming techniques, contextual inquiry methods, physical prototypes, and more. These tools aid designers in their creative and problem-solving processes, and they may evolve over time as new design methods and technologies emerge [71]. Across the reviewed studies, AI tools have been implemented in several distinct contexts. In electronics education, a debugging tool has been developed to support students in completing electronic design assignments [30]. In machine learning education, a web-based tool has been designed to explain classification concepts to beginners without requiring technical expertise [52]. The Scratch programming environment has been adopted to introduce AI concepts to students aged 16–18 [51]. Visualisation tools have also been developed for teaching computer security concepts [31], while playful authoring tools have been implemented to support users in creating interactive experiences [36]. In engineering

https://doi.org/10.3390/ai7050160

education, a sketch recognition tool uses AI to detect and analyse the shapes and features of free-body diagrams [44]. Finally, in creative design, a tool employing multimodal inputs such as images and colours has been developed to help designers explore and express their intentions to AI [58].

6. AI Techniques for Implementing Intelligent Behaviours

6.1. Understanding the Storage Knowledge and Updating

Over the last twenty years, the growth of the Semantic Web has enabled the emergence of a large selection of structured data on the internet, presented as knowledge bases [72]. A knowledge base (KB) is defined as a structured database that includes groups of facts [73]. Researchers have used different ways to store knowledge. The KB used is static and only updated in the small talk database, in case the chatbot does not understand the text entered by the user. The dialogue manager is a response to provide an appropriate answer to the user by matching the input with KB. The semantic network in [57] built from 1255 Kansei words, by using data from consumer comments and automotive critic reviews. These words are then mapped to car models to provide the relevant design features.

Coding Tutor has three different databases to support students in learning [41]. The Learning Path Object database is built to store information related to the progress and performance of learners. Independent Learning Object is used to store concepts related to homework and activities. A Small Talk Response database can be used if the student's intent is not typically related. The dataset collected in [46] consists of hand-written sketches.

The database in May AI [37] is dynamic. The feedback from the designer is used to update the probability distribution of each suggestion agent. The COFFEE bot [22] used a database to store course content and an internal database to log user interactions.

Recommender systems make use of ontologies as the back end of the system to represent knowledge and simplify the parsing, reasoning, sharing, and reuse of knowledge [45]. This system of organising data into distinct concepts with associated attributes and relationships helps to create more precise recommendations for students. The ontology includes information about higher education, students, and employment.

6.2. Algorithms and Implementation

An algorithm is defined as, "the thing that gets data processing and other computation done", based on [74]. This section discusses some existing algorithms of papers mentioned in Section 4. The implementation of different approaches in AI design tools is presented in Table 3. Our taxonomy is summarised in Table 4.

6.2.1. Conversational AI

The chatbot in [41] is implemented based on natural language processing, which incorporates natural language understanding to generate input to the dialogue manager and knowledge base (KB) if the user's intent is matched with the words stored in the database. Additionally, natural language generation is used to show a response to the user accurately. An intelligent personal assistant [26] was implemented on mobile devices, allowing users to interact through a chat window. The chatbot is implemented using the Dialogflow platform for natural language processing and acquiring input from the user. The chatbot described in [22] was implemented using Dialogflow along with a learning framework to customise course material for individual student performance.

The virtual teaching assistant (VTA) was implemented using Rasa in [21]. RASA is free and open-source for creating virtual assistant (AI) software [21]. A machine algorithm was applied to build Rasa, and the NLU model was used and able to understand the intent of

https://doi.org/10.3390/ai7050160

the student's input. The interaction between students and the system is based on messages, and the RASA core interprets these messages.

Table 3. Implementation of different approaches in AI techniques, a dash (–) is used to indicate that the information was not reported, All abbreviations used in this table are explained in Table 4.

Names of System/Tool	**Publication Year**	**Knowledge Store**	**Knowledge Update**	**Entering Input**	**Algorithm/Method**	**Interaction Types**
(1)	(2)	(3)	(4)	(5)	(6)	(7)
MetaMorph [46]	2020	DS	—	D	DNNs	V
ImageSense [38]	2020	DB	—	D	NLA, MMCQ	W,V
May AI [37]	2019	DB	D	U	CCB	V
iList [42]	2015	—	D	U	ML	W,V
Assistance System [23]	2022	DB	D	CH	IR	V
ICICLE [43]	2000	KB	D	U	NLP	W
Coding Tutor [41]	2019	DB	D	NP	—	W
Alexa for E-learning [49]	2020	—	—	U	LAM, NLA	S
An Intelligent Reading Assistant System [32]	2021	DB	—	L	RA	W
Intelligent English Teaching Platform [25]	2021	NB	—	I,L	DTA, NN	W
The Toastboard [30]	2016	—	—	U	—	A
Teachable Machine [52]	2020	—	—	U	ML	V
Zhorai [53]	2020	—	—	CH	—	W,V
A Scratch-based AI tutorial [51]	2019	—	—	L	CL, NN	V
Smartpathk [33]	2021	—	D	I,L	SML, J48	W,V
Chatbot [26]	2018	—	—	L	—	W
An Intelligent Assistant [20]	2015	—	—	L	—	W
Visualisation Tools [31]	2010	—	—	L	—	W,V
VTK [35]	2006	—	—	L	—	W,V
Playful Authoring Tools [36]	2017	—	—	U	—	W,V,S
Mental Vision [34]	2006	—	—	L	—	W,V
Mechanix [44]	2014	—	—	L	GBA, RTruss	W,V
Trigger-action-circuits [24]	2017	DB	—	U	BFS, Rec, DRA	A
CVTA [21]	2020	—	—	L	ML	W
HeyTeddy [28]	2019	—	D	NP	—	W,S,A

Table 3. *Cont.*

Names of System/Tool	Publication Year	Knowledge Store	Knowledge Update	Entering Input	Algorithm/Method	Interaction Types
ThinkInk [29]	2018	—	—	U	RATA	W,V
CSTutor [47]	2014	—	—	L	Gesture recognition	W,V
Smart Personal Assistant (SPA) [50]	2019	—	—	U	NLP	S
AutoSpark [57]	2024	NB	S	D	KE, LLM, PatchInv, CLIP	W,A,V
LearningML [54]	2020	DS	D	L	ANN, supervised learning	W,S
COFFEE [22]	2024	DB	D	L	NLP and ChatGPT	W,A,V
DesignPrompt [58]	2024	—	—	U	OpenAI	W,A,V
Q-Module-Bot [27]	2024	KB	D	L	Web-scraping and GPT-3.5	W,A

Table 4. Summarised taxonomy used in Table 3, starting from the second column, as the first column is already included in Table 3.

	Taxonomy	Description
3	Knowledge store	Database (DB), Knowledge Base (KB), Dataset (DS), Text Files (TF), Folder (FO)
4	Knowledge update	Static (S), Dynamic (D)
5	Entering input	Patients (P), Users (U), Learners (L), Novice Programmers (NP), Elderly People (EP), Instructors (I), Children (CH), Designer (D), Pregnant Women (PW)
6	Algorithms or methods	Decision Tree Algorithm (DTA), Machine Learning (ML), Generate Guidelines and Monitor Health (GG and MH), Natural Language Algorithm (NLA), Recommendation Algorithm (RA), Deep Neural Networks (DNNs), Modified Median Cut Quantisation (MMCQ), Cooperative Contextual Bandits (CCB), Image Recognition (IR), Breadth First Search (BFS), Dependency Resolution Algorithm (DRA), Supervised Machine Learning (SML), Neural Network (NN), Graph-Building Algorithm (GBA), Recognition and Translation Algorithm (RATA), Clustering (CL), Recognising Truss (RTruss), Recursive (Rec), Lambda (LAM), Kansei Engineering (KE), Large Language Model (LLM), PatchInv (Patch-inversion), Contrastive Language–Image Pretraining (CLIP), Chat Generative Pretrained Transformer (ChatGPT), Open Artificial Intelligence (OpenAI)
7	Types of interaction	Written (W), Spoken (S), Action (A), Visualisation (V)

6.2.2. Systems

An ontology-based recommender system integrated with machine learning has been proposed to help students choose their major after high school [45]. Data is collected through student profiles and surveys to identify individual interests. Machine learning algorithms, including K-mode, self-organising map, and hierarchical clustering, are then applied to filter and cluster the data before generating personalised recommendations.

By leveraging the intelligent reading assistant system proposed in [32], educators can upload text articles which are automatically converted into a knowledge graph by the system. Furthermore, two methods are employed for generating test questions: one based on the knowledge graph and the other one utilising Recurrent Neural Networks (RNNs) for text generation. Such approaches allow learners to develop their cognitive abilities in reading and answering test questions accurately. Additionally, the system can generate reports based on individual learner profiles, thus providing personalised learning paths for improvement.

Trigger Action Circuits is a powerful software application that runs on a computer and is implemented using the C# programming language [24]. It defines components using an XML file and allows users to easily create and modify them. Furthermore, the implementation uses a breadth-first, recursive, dependency resolution algorithm to generate circuits that satisfy the mapping. This algorithm looks for possible solutions for each component and keeps track of components that have been used or are available. AI techniques such as decision tree algorithms and neural networks have been leveraged to design a platform for teaching English [25].

Researchers in [33] used the J48 algorithm to implement the SmartPathK platform. The algorithm, which is part of the machine learning family, enables a computer to model a decision tree, making it possible to make decisions based on certain conditions. The accuracy of the tool was tested and found to have an 89.47% accuracy when using the machine learning algorithms based on the decision tree. Data entered by a user is stored in the decision tree algorithm, and the knowledge base created by the J48 algorithm is not static, as it can be updated with new data. This allows for more accurate decision-making based on specific conditions. The J48 algorithm was validated by testing its accuracy in identifying glomerulopathies.

The implementation of the intelligent tutoring system [42] was enhanced by utilising machine learning approaches in order to create an effective model based on the past experiences of a student engaging with the system. The procedural knowledge model was built to automatically generate a useful model from the student's previous interactions with the system. This model is represented by a directed graph having two types of vertices: states and actions. States signify a snapshot of the system, while the actions depict the probabilities of a student taking a certain action. This model is used to assess the student's uncertainty, and the graph is traversed to estimate the probability of the student reaching a correct solution.

The AutoSpark system was implemented by merging several techniques to improve the design process [57]. The Kansei Engineering engine combines outputs from large language models such as GPT-4/GPT-4V, with a semantic network of Kansei words to recommend design features. Moreover, it uses a patch-inversion method to highlight how parts of a generated image respond to text prompts. Also, AutoSpark uses CLIP-based comparisons to guarantee the image matches the intended description.

6.2.3. Tools

Researchers implemented two algorithms, K-Means and an Artificial Neural Network logic gate, which were coded into a Scratch template as detailed in [51]. This template is designed with empty blocks, which users need to fill to execute the algorithms. For the K-Means algorithm, students need to complete the specification of a block called KMeans in order to calculate the distance from each of the N points to K mass centres and allocate each point to the nearest mass centre. An Artificial Neural Network logic gate implements the algorithm; the students must code the equation to update the weight of the network, with the equation for another weight already provided in the Scratch file. After the code is completed, students can train the neuron and observe the evolution of weights and output errors.

Researchers implemented Mechanix, an intuitive, natural sketch interface, to aid novice software students in learning truss analysis and diagrams [44]. This tool allows students to draw trusses and diagrams more intuitively with a pen-like interface. The interface also provides feedback to the student on the correctness of their diagrams, with colour-coding and arrows to highlight errors and provide information to aid the user in understanding and correcting them. To identify truss shapes from multiple line shapes, researchers used a graph-building algorithm. Furthermore, AI algorithms were also utilised to recognise and compare trusses. The system has a built-in voice recognition module that converts a user's sound into text [49]. A natural language understanding (NLU) module then analyses the text. A keyword matching module is used to recognise, respond to, and adapt to different scenarios depending on the words used by the learner. To store and analyse data, the learning system was developed by Amazon Web Services (AWS) lambda functions, which stores the data in Amazon S3. Additionally, Amazon Comprehend is used to further analyse the data and extract meaningful insights.

6.2.4. AI Assistants

A supervised deep learning model was used to create MetaMorph, enabling the model to identify and classify collected datasets and generate high-level sketches with better accuracy [46]. A related web-based implementation utilises common web technologies such as HTML, CSS, and JavaScript for the user interface, with a Python-based back end [38]. Google vision API is used to retrieve semantic labels of images, attaching the top ten labels when a designer adds a new image to the board. Additionally, the MMCQ algorithm is applied to analyse each image for its ten major colours. A machine learning framework based on the cooperative contextual bandits (CCB) algorithm has been adopted to offer relevant pictures or text to users [37]. The algorithm makes decisions based on expected probabilities for each option. To optimise suggestions, the agent either exploits its current suggestion if it provides the highest probability, or explores other options by referring to another agent with a higher probability.

Before discussing the evaluation of AI techniques, Table 3 is examined temporally to identify how implementation approaches have evolved across the reviewed period. During the early period (2000–2015), techniques such as [31,34,35,42–44,47] did not report a knowledge store or knowledge update mechanism, and targeted learners as the primary input type. Algorithms were largely absent from reporting, and interaction was mostly written or visual. During the middle period (2016–2020), database-backed knowledge stores became more common, as seen in [37,38,41], and dynamic knowledge updating began to appear. Furthermore, this period introduced more diverse algorithmic methods, including deep neural networks, cooperative contextual bandits, and supervised machine learning. Voice and action-based interactions also emerged through systems such as [28,49]. In the most recent period (2021–2024), systems such as [22,27,57] integrated large language models and generative AI APIs, supported dynamic knowledge updating, and combined multiple interaction modalities simultaneously.

6.3. Evaluation

In this section, we thoroughly examine the evaluation methods used in each paper to assess AI techniques. We focus specifically on the question: How do they evaluate? As indicated in Appendix A Table A1, most papers used experimental evaluation methods.

Furthermore, various approaches were used for data collection across chatbots, AI systems, AI tools, and AI assistants. This table also discusses the specific metrics targeted by each paper during their evaluation, either derived from user feedback or the techniques themselves. In Table 5, we categorised the AI techniques to analyse the implementation of experimental evaluation. Our analysis indicates that the majority of AI techniques incorporated both quantitative and qualitative methods.

Table 5. Classification of AI techniques based on their employed methodology: qualitative, quantitative, and mixed approaches.

	Chatbot	**System**	**Tools**	**AI Assistance**
Qualitative	Elragal et al. [22]			Konecki et al. [20], Winkler et al. [50], Villegas-Ch et al. [23], Koch et al. [38]
Quantitative	Allen et al. [27]			
Mixed	Pham et al. [26], Kim et al. [28]	Hobert [41], Anderson et al. [24], Chen et al. [57], Sun et al. [25], Fossati et al. [42], García et al. [54]	Drew et al. [30], Yuan et al. [31], Estevez et al. [51], Atilola et al. [44], Imtiaz et al. [29], Peng et al. [58], Buchanan and Laviola Jr [47]	Sermuga Pandian et al. [46], Koch et al. [37], Rodríguez-García et al. [55]

Evaluation Challenges and Limitations in AI Techniques: This section explores the common challenges and limitations faced by other researchers. We also suggest some solutions to help overcome these limitations.

Challenges in Engagement and User Interaction: Engagement and user interaction remain common issues in several AI techniques, revealing the underlying difficulties faced by users across various systems and applications. For example, regarding chatbots [26], users expressed limitations and dissatisfaction with a chatbot's responses, further demonstrating how a complex or unintuitive interaction design can hinder user engagement. In [37], some users found it challenging to understand AI's suggestions in the design process, and concerns were raised about how AI could lead to laziness in design thinking. Even in educational settings, as seen in AI assistance [50], challenges adapting to and effectively utilising tutoring systems were reported, indicating that these barriers are not confined to one area of technology but extend across various domains.

To mitigate these challenges, it may be helpful to conduct iterative design and testing that vigorously involves end users early in the development process. For example, qualitative feedback, such as interviews and usability tests, can be used to improve interactions and guarantee that the system adapts to user expectations. In addition, developing text-based systems by integrating voice, touch, or visual inputs can provide more rich interactions and better accommodate different user preferences.

Limitations in Technology and Models: Several AI techniques exhibit a consistent pattern of technological limitations and model constraints, highlighting widespread difficulties in accurately and effectively applying different systems. In physical computing education, challenges have been identified in handling complex prototypes, the lack of multi-user support, and difficulties in integrating new features [28]. In programming education, diverse user experience levels, the need for software improvements, and server capacity limitations reflect struggles with adaptability and responsiveness [41]. Further challenges include limited support for higher-level software constructs and difficulties in supporting circuits without a microcontroller [24]. Finally, constraints in handling complex circuit debugging scenarios, including a limited ability to capture rapidly changing values and analyse embedded software behaviour, indicate the need for new solutions [30].

One solution this paper suggests for enhancing the model is continuous model evaluation. This could be achieved by implementing robust monitoring and by using both quantitative and qualitative evaluation to ensure sustained performance and accuracy, even with evolving user inputs.

Challenges in AI Education and Learning Techniques: The study in [42] emphasises dissatisfaction with repetitive feedback in a specific version of iList, linking it with the overall theme of user experience and feedback. References [29,31,51,53,55] share a common concern regarding small sample sizes, limiting the strength and generalisability of conclusions, and point to the need for more extensive research and controlled interaction. This reflects a broader challenge in educational research, as mentioned in [50], where adaptation to new tools like the SPA tutor may present limitations compared to human interactions. The researchers in [23] touch on user experience again, this time focusing on the unique challenges faced by young children in terms of technical complexity and cognitive load.

The solution could be that future studies incorporate larger, more diverse participant pools and evaluation processes to capture long-term user interactions and learning outcomes. Additionally, to reduce the cognitive load and complexity that may face participants, before developing AI techniques, the aimed user should be incorporated into the design of the techniques, such as by using a co-design approach [75]. The co-design approach may help to understand what they need and what the level of their knowledge is. Based on the results of this approach, the technique could reduce complexity and cognitive load for learners by designing a technique that suits them.

Limited Sample Size Affecting Conclusions: Some challenges identified in [53] raise concerns regarding the evaluation of Zhorai's effectiveness. The study was conducted with a small number of participants, indicating that further testing is needed to determine whether Zhorai is effective. The results of the student questionnaires and educational visualisation tools are promising in [31], but the limited sample size and one-time workshop make it challenging to draw definitive conclusions, emphasising the need for more extensive studies, sophisticated materials, and alternative programming languages [51].

In addition, limited availability, potential bias in selection, and curriculum changes present challenges in evaluation [44]. Individual biases and interpretations, the time-consuming nature of data collection, and the possibility of socially desirable answers instead of true feelings reveal the inherent complexities in the methodology [29]. Reductions in the sample size, potential influence of prior knowledge, and gender disparities in STEM may also affect the generalisability of the findings [29]. Finally, the challenges for users, such as adapting to new tools and methodologies, encapsulate the multifaceted nature of these challenges [50].

User Dissatisfaction and Experience Challenges: Users were dissatisfied with the app functions and chatbot interactions [26], illustrating the critical need for a user-centric design. Feedback issues caused dissatisfaction among students, with one version being viewed as more repetitive and misleading than others [42], emphasising the importance of clear, varied feedback. Users also struggled with understanding an AI system's suggestions, leading to concerns about its complexity and relevance [37]. The challenges with adapting to an automated tutor, compared to a human tutor [50], demonstrate potential limitations in the technology's ability to mimic human-like understanding and responsiveness. The solution could be conducting a pilot study of periodic usability sessions with diverse groups of participants to gather targeted feedback on specific AI techniques. This can help identify and correct development issues before launching products.

Complexity and Difficulties in Understanding Systems: The ideas highlighted underscore a prevalent theme of complexity and difficulties in understanding various systems, models, or technologies across different contexts. Design limitations exist in handling prototypes [28], debugging difficulties [24], and addressing complexities in circuit debugging [30]. Learning curves and technical limitations in the evaluation process [46], user understanding of AI-driven processes [37], and technical complexity in children's interactions [23] further emphasise the multifaceted nature of these challenges. Whether

https://doi.org/10.3390/ai7050160

in medical contexts, user interface design, age-appropriate system design, or technology evaluation, these complexities underline the need for a user-centric and context-aware approach. They highlight how intricate details, lack of user-friendly design, and inherent complexities can lead to implementation challenges, underscoring the importance of multi-faceted solutions that consider users' needs, technical capabilities, and the specific context in which the system operates.

6.4. Types of User Interactions

Regarding the ways in which users interact with AI techniques, the interaction can be divided into four types: text, voice, visualisation, and action. These allow users to access machines and receive accurate responses to their queries. The following section discusses these types in detail.

6.4.1. Interaction Based on Writing

Text-based interaction is a form of communication between the end user and an AI techniques, where the user interacts with the AI system through written text. This form of communication makes it possible for users to interact with AI techniques in real time, allowing for more natural and effective conversations. Interactions based on text can be used to help users receive information, request services, give feedback, and more. In addition, this type of communication can also be used to help AI systems learn and improve their understanding of the user's needs.

6.4.2. Interaction Based on Visualisation

The use of visualisation in the interaction between users and AI techniques has numerous benefits. Visualisation helps to quickly convey complex information to users in an way that is easier to understand. Visualisation of the output helps speed up the processes of understanding and decision-making, allowing users to understand the output of AI tools quickly and accurately. An evaluation of the effectiveness of Zhorai was conducted with 14 children in small groups [53], which revealed an increase in engagement during learning due to the conversational aspect. Using a combination of conversation and visuals was successful in educating kids on the fundamentals of machine learning. The use of visualisation in the interaction between users and AI tools can help to reduce cognitive overload and improve the user experience, making the interaction more enjoyable and efficient.

6.4.3. Interaction Based on Voice

Learning from AI technology is not limited to text-based interactions or visualisations; instead, it has been further augmented to include voice interactions to meet the needs of users. Voice assistants support a range of activities, including responding to user questions, creating alarms, making calls, and providing weather updates [76]. In educational contexts, voice-over interaction has been shown to offer advantages over typing, particularly in making programming more accessible for novice learners [28]. Voice-based assistants, particularly Alexa, have been adopted across several reviewed studies [28,49,50]. An Amazon Echo Dot device is used to handle voice input and convert it to text [28], while Alexa has also been applied as an intelligent personal assistant for English language education [56].

6.4.4. Interaction Based on Action

By actually performing tasks and seeing the results in real-time execution, users can observe how AI can help them solve problems more quickly and efficiently. Additionally, the ability to test out different scenarios and observe the results can provide valuable insight into how AI techniques can be used to improve existing processes and create new ones. As noted in Section 4, beginner programmers can learn by building and assembling circuits [24,28,30]. AutoSpark combines various interaction modalities [57]. Designers can type text prompts and Kansei words, click and select options, upload images, and view visual representations on the design board.

Table 6 summarises the interaction types across the reviewed studies. Interaction types were coded using a multi-label approach, as some of AI techniques support multiple modalities simultaneously (e.g., text, voice, action, and visualisation); see Table 3. Each study could be assigned multiple interaction labels; therefore, counts reflect the number of studies in which each interaction type appears rather than mutually exclusive categories. Percentages reported in Table 6 are calculated by normalising the total number of interaction occurrences across all studies. Written interaction was observed in 25 studies, followed by visualisation in 19 studies, action-based interaction in 7 studies, and spoken interaction in 5 studies. When normalised across the total number of interaction instances, written interaction was the most frequently observed (45%), followed by visualisation (34%), action-based interaction (12%), and spoken interaction (9%). These proportions are not mutually exclusive, as individual studies may include more than one interaction type.

Table 6. Interaction types (text, spoken, action, and visualisations) in academic research papers.

Interaction Types	Number of Papers
Written	45%
Spoken	9%
Action	12%
Visualisation	34%

6.5. Generative AI and Large Language Models in Educational Assistants

As discussed in this review, a substantial proportion of identified AI educational systems, e.g., rule-based logic, machine learning, and neural networks, often rely on predefined rules and a static knowledge base. However, although these techniques provide reliability and control, their interaction capabilities with learners are often constrained by predefined workflows, restricting flexibility and constrained learning.

On the other hand, large language models (LLMs) and generative AI (GenAI) systems represent a paradigm shift in AI-assisted learning [2]. The systems reviewed, such as DesignPrompt, COFFEE, AutoSpark, and Q-Module-Bot, integrate LLM-based instruments (e.g., ChatGPT, GPT-3.5, and OpenAI APIs) to dynamically generate responses. In contrast with rule-based techniques, LLMs generate explanations and feedback from large-scale pretrained knowledge rather than manually encoded rules. This allows more adaptable and conversational forms of learner interaction that can respond to diverse inputs and contexts. Therefore, LLM-based systems are increasingly positioned as promising techniques to help facilitate meaningful learning experiences.

The reviewed studies present some models for integrating LLM into education. Some systems directly integrate OpenAI or GPT-based APIs to generate feedback design suggestions or answers to user questions. In these cases, the LLM acts as the immediate reasoning engine producing output based on user instructions. Some systems combine traditional knowledge representations (such as knowledge bases ontology or datasets) with generative models. For example, web scraping combined with GPT-3.5 or multimodal models such as CLIP allows systems to map responses to structured inputs while leveraging generative flexibility. Systems such as AutoSpark include multimedia capabilities (such as image-to-text integration via CLIP) along with LLM logic. This enables learning through visual, textual, and interactive channels, going beyond purely textual platforms. These approaches reflect a transition away from the enforcement of fixed rules towards generative and context-aware response mechanisms. Integration of LLMs has significant implications for the explicit and implicit learning features identified in Section 4. LLM-based assistants support explicit learning by enabling explanations of complex concepts, guided questions, tutorials, and adaptive feedback tailored to the learner's responses. As LLMs can generate dynamically different responses, they assist learners in engaging in an iterative dialogue rather than receiving static feedback. This could increase the depth of conceptual engagement and support reflective learning processes. For instance, when a learner poses a question to an LLM-based assistant, the generated response may encourage further inquiry, fostering iterative engagement that supports the development of critical thinking and curiosity. Generative systems also support implicit learning features through the recommendations and learning-by-experimentation within conversational systems. Learning happens indirectly as users explore system outputs, change questions, and observe the results. Thus, LLM integration boosts both explicit and implicit learning by increasing flexibility, responsiveness, and contextual adaptation.

Beyond the findings of the included studies, several broader considerations are relevant to the design and evaluation of LLM-based educational assistants. Although LLM-based systems expand the capabilities of AI assistants in education, they also introduce new evaluation and ethical challenges. Traditional evaluation metrics of AI focus on metrics such as accuracy, task completion rates, and usability scores. Those may be insufficient for generative systems. Thus, additional evaluation dimensions are necessary, including the accuracy and reliability of feedback responses [77], as LLM-based educational systems may present risks to learners, such as hallucination. This could lead to reduced critical thinking if learners accept outputs. These risks are especially critical in some fields such as privacy, security, and health, as incorrect guidance may have significant effects on learners. Therefore, the integration of LLMs into educational assistants requires careful design of guardrails and evaluation.

7. Research Challenges and Directions

This section outlines four primary research challenges identified through the literature review. While AI educational assistants have demonstrated growing capabilities in supporting learning, significant technical and evaluative limitations persist.

7.1. Conversational Adaptivity and Interaction Limitations

Natural language processing and generative models have enabled more interactive learner engagement according to the review. However, many of the systems reviewed still struggle to support both explicit and implicit learning through adaptive interaction. Many AI educational systems in the review lack support for conversational interaction, personalised feedback, and mechanisms for iterative dialogue. Rule-based and machine learning approaches are particularly limited in handling open-ended learner queries or dynamically providing explanations. As a result, the interactions implemented in many techniques remain limited. To address these constraints, future educational assistants should incorporate mechanisms that support adaptive dialogue, guided questioning, and context-aware feedback. Large language model (LLM)-based systems offer one potential direction, particularly when designed to provide conversational flexibility with structured learning support.

7.2. Knowledge Representation and Architectural Constraints

This review also identified differences in how knowledge representation is implemented and documented. A considerable number of studies did not specify whether their

https://doi.org/10.3390/ai7050160

knowledge base was static or dynamically updated. As noted, several studies relied on static knowledge bases without update mechanisms, thereby limiting adaptability to developing fields. Moreover, systems built on rule-based logic or traditional machine learning models are often unable to adapt to open-ended learner engagements. This restriction limits the flexibility and depth of the learning support they provide for learners. Future systems should adopt updatable architectures for separate knowledge representation from interaction logic. Such separation allows knowledge bases to operate independently without requiring retraining of the system. Retrieval-augmented generation (RAG) [78] represents one approach for grounding generative outputs in verified knowledge sources. This thereby improves reliability while supporting adaptability. These design directions may contribute to more scalable and robust educational assistants.

7.3. Integrating Heterogeneous AI Approaches

Most reviewed systems are based on a single dominant technique, such as conversational agents [22,27], teaching platforms [26,34] or intelligent tutoring systems [47,68]. While each approach offers advantages, many systems lack the capability to integrate multiple learning support techniques. Deep learning support requires generating structured explanations, responding to questions, and supporting iterative interaction. However, rule-based or narrowly trained machine learning systems frequently struggle to support these functions comprehensively. Future studies should investigate hybrid architectures that combine complementary techniques, such as integrating machine learning components with LLM-based conversational layers. These heterogeneous designs may improve alignment of implementation mechanisms with the different interaction features identified in the review.

7.4. The Need for Robust Evaluation Frameworks

A finding in the reviewed literature is the unevenness of evaluation practices. As summarised in Appendix A, Table A1, studies vary substantially in their designs, sample sizes, durations, and reported metrics. Many rely on short-term experiments or single-session workshops. Thus, it limits conclusions about the long-term impact on learning. To address this, more rigorous evaluation designs are necessary. Longitudinal studies that track learners' knowledge, behaviour, and skill development over long durations, such as a full academic semester, would provide stronger evidence of sustained learning effects. The 12-week evaluation conducted for Q-Module-Bot [27] provides an example of a more extensive evaluation. Evaluation approaches should also extend beyond usability and effectiveness metrics to include learning-specific measures such as knowledge gain and misconception analysis. Regarding systems built with LLMs, evaluation pipelines should include expert review, automated verification against trusted knowledge sources, and the accuracy of answers that LLMs produce. Thus, reflect transparency mechanisms for support engagement with AI-generated responses.

7.5. Limitations

This review has several limitations that should be considered when interpreting the findings. The first limitation is related to the fact that the presented review was limited to English-language publications only. This could have resulted in the exclusion of relevant studies published in other languages, which could have resulted in a language bias. Second, the classification of studies as explicit or implicit learning was subjective, based on the theoretical definitions of explicit and implicit learning drawn from educational theory. Future work could develop a classification framework for AI educational techniques, which could then serve as a reference guide for future research in this field. Finally, only four studies among the total 38 studies included in the present review incorporate the use of

LLMs or generative AI. Therefore, this means that the proposed taxonomy and findings are focused mostly on the pre-period of LLM expansion in learning. A future review could focus on the educational techniques based on LLMs only to propose a more representative taxonomy for understanding the different mechanisms underlying the learning process in the context of the latest AI-based educational assistants.

8. Conclusions

This study examined how AI educational systems support learning through interaction features. We analysed 38 studies and proposed a taxonomy of AI techniques that support learning features through different mechanisms. The review identified that explicit learning is supported by 11 interaction features, such as asking questions and providing feedback. Implicit learning is supported through four features including asking and answering questions, performing activities, and feedback. The findings show that 79% of reviewed studies supported explicit learning and 21% supported implicit learning. In addition, we analysed their implementation approaches by examining knowledge representation strategies, algorithms, and interaction mechanisms. The findings indicate that four implementation approaches were identified: conversational AI, AI systems, AI assistance, and AI tools. Algorithms evolved from rule-based and machine learning methods toward large language models in the most recent period. Written interaction dominated (45%), followed by visualisation (34%), action (12%), and voice (9%). Most studies relied on experimental evaluation combining qualitative and quantitative methods. We also discussed evaluations of AI techniques and identified the challenges and limitations of AI techniques. We highlighted limitations that remain in conversational adaptivity, knowledge representation, and evaluation. We identified four research gaps and highlighted the need for more adaptive architectures, clearer reporting standards, heterogeneous integration strategies, and more rigorous evaluation frameworks. Overall, this review provides a structured foundation for analysing how interaction features are implemented and evaluated in AI educational assistants, offering guidance for developing more robust and adaptable learning systems.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/ai7050160/s1.

Author Contributions: Conceptualisation, F.A. and C.P.; methodology, F.A., C.P. and N.A.; formal analysis, F.A.; data curation, F.A.; writing—original draft preparation, F.A.; writing—review and editing, F.A., C.P. and N.A.; review, O.R.; supervision, C.P. and O.R.; funding acquisition, C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by EPSRC PETRAS (EP/S035362/1).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data generated during this study consist of extracted and coded information from the included studies. These data include the list of included papers and extracted variables (e.g., learning type, implementation approaches, and evaluation characteristics). All materials are provided as Supplementary Materials.

Acknowledgments: The authors would like to acknowledge the scholarship and support provided by Najran University, Najran, Saudi Arabia.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Evaluation of AI approaches: chatbots denoted by beige background, AI systems by blue, AI tools by green, and AI assistants by grey. A checkmark (✓) indicates that the study supports the corresponding evaluation approach, while a dash (–) indicates that no findings were reported.

Ref.	Eval. Types		Data Collection Methods	Participants	Metrics/Measures
	Exp	Theo			
Pham et al. [26]	✓		-	14,000 people	User–chatbot interaction and usability.
Kim et al. [28]	✓		Sessions with participants and interviews	10 users	Time, debugging issues, and users' confidence
Shekhar et al. [21]	-	-	-	-	-
Elragal et al. [22]	✓		Workshop demonstration and qualitative feedback	learners, instructors, and academic admins	Perceived feasibility, usability observations, and collecting qualitative feedback
Allen et al. [27]	✓		Questionnaires (surveys)	5 academic stakeholders + 5 M.Sc. holders	usability, confidence, extensibility, and relevance
Fossati et al. [42]	✓		Tests for learning assessments, surveys, and student action records	between 214 and 219 students	Learning, satisfaction, and problem-solving behaviour
Hobert [41]	✓		Demonstrations, self assessments, programming tasks, questionnaires, and written feedback	40 students	Usability, effectiveness, efficiency, and acceptance among its intended users
Michaud et al. [43]	-	-	-	-	-
Zhao et al. [32]		✓	-	-	Reading ability
Anderson et al. [24]	✓		Complete a task: building a circuit using an Arduino with specific time points	12 participants	Usability, time, and efficiency of the system
Aldeman et al. [33]	-		-	-	The accuracy of the model generated, the comprehensive ability of the extracted knowledge, and the learning time
Zhao et al. [49]	-	-	-	15 learners	The improvement in learners' interviewing skills after using the application
Chen et al. [57]	✓		Questionnaires and semi-structured interviews	16 professional designers, 50 consumers, and 4 expert reviewers	Emotional expression, aesthetic quality, and novelty of designs, user ratings (match, transparency, and ease of use), NASA-TLX subscales (mental demand, effort, and frustration)
Sun et al. [25]	✓		-	20–120	Learning activity participation, Interaction level, resource utilisation quality of interaction, efficiency, and student performance
García et al. [54]	✓		Questionnaires (pre-test and post-test), including Likert questions and open questions	14 students	Pre-test and post-test scores, and improvements in AI knowledge
Drew et al. [30]	-	-	Interviews and observations	7 participants	Learning gain (pre-/post-tests), problems solved, feedback, satisfaction survey (Likert + qualitative comments)
Lin et al. [53]	-	-	Pilot Study and user study	14 children	The effective engagement of Zhorai and the children's understanding
Yuan et al. [31]	✓		A combination of pre-tests, post-tests, anonymous questionnaires, and surveys	15 students	The usability and effectiveness of tools in various computer network and information security courses
Estevez et al. [51]	✓		Questionnaires and open questions	37 students	Understanding AI
Carney et al. [52]	-	-	-	-	-
Atilola et al. [44]	✓		Assessment and focus groups	70–100 regular students, 30–40 honors students	The effectiveness of Mechanix
Peternier et al. [34]	-	-	-	-	-
Imtiaz et al. [29]	✓		Likert-scale questionnaire	45 students	Interactivity, design, playfulness, ease of use, usefulness, and intention to use
Dias et al. [35]	-	-	-	-	-
Peng et al. [58]	✓		Structured observation of design tasks, think-aloud protocol, semi-structured interviews, likert-scale questionnaires	12 professional designers	Effectiveness of multimodal inputs in capturing creative intentions, understanding the mapping between inputs and AI outputs, sense of control and transparency over the creative process, satisfaction and usability of the system
Buchanan and Laviola Jr [47]	✓		Pre-test quizzes, surveys, and course exams	190 students (from an initial 267)	Pre-test and quiz scores, Likert scale surveys, and comments

https://doi.org/10.3390/ai7050160

Table A1. *Cont.*

Ref.	Eval. Types		Data Collection Methods	Participants	Metrics/Measures
	Exp	Theo			
Koch et al. [37]	✓		Standardised measurements and semi-structured interviews	16 professional designers	The effectiveness and impact of an AI tool on creative design
Villegas-Ch et al. [23]	✓		Assessing children's progress in learning numbers	Group of six children	The progress and performance of children in learning numbers
Konecki et al. [20]	✓		Questionnaire	68 students	Students' programming motivation, intelligent assistant's impact on time investment, assistant's information retrieval efficiency, perceived usefulness of assistant, preference for the method, and improved programming understanding
Koch et al. [38]	✓		Observation, interviews, questionnaire screening, and audio recording	9 designers	-
Rodríguez-García et al. [55]	✓		Multiple-choice questions and pen questions	494 children	knowledge improvement, perception of learning ML, and changes in the perception of AI.
Sermuga Pandian et al. [46]	✓		ASQ questionnaire, semi-structured interviews, think-aloud protocol, and task recording	10 designers	ASQ scores (Q1–Q3)
Winkler et al. [50]	✓		Assessing task outcomes, ensuring collaboration quality, and analysing video recordings	63 participants	Assess whether the AI tutor can improve task outcomes and collaboration quality compared to human tutors in collaborative problem-solving settings

References

1. Sequeira, A.H. Introduction to Concepts of Teaching and Learning. Available at SSRN 2150166. 2012. Available online: https://www.researchgate.net/publication/272620585_Introduction_to_Concepts_of_Teaching_and_Learning (accessed on 15 April 2026).
2. Chen, L.; Chen, P.; Lin, Z. Artificial Intelligence in Education: A Review. *IEEE Access* **2020**, *8*, 75264–75278. [CrossRef]
3. Seo, K.; Tang, J.; Roll, I.; Fels, S.; Yoon, D. The impact of artificial intelligence on learner–instructor interaction in online learning. *Int. J. Educ. Technol. High. Educ.* **2021**, *18*, 54. [CrossRef] [PubMed]
4. Wang, D.; Huang, X. Transforming education through artificial intelligence and immersive technologies: Enhancing learning experiences. *Interact. Learn. Environ.* **2025**, *33*, 4546–4565. [CrossRef]
5. Chen, Y.; Jensen, S.; Albert, L.J.; Gupta, S.; Lee, T. Artificial intelligence (AI) student assistants in the classroom: Designing chatbots to support student success. *Inf. Syst. Front.* **2023**, *25*, 161–182. [CrossRef]
6. Suh, S.; Ravelo, J.; Strogalev, N. Impact of Artificial Intelligence on Student's Education. *J. Comput. Sci. Coll.* **2025**, *40*, 80–90.
7. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. [CrossRef]
8. Kitchenham, B.; Brereton, P. A systematic review of systematic review process research in software engineering. *Inf. Softw. Technol.* **2013**, *55*, 2049–2075. [CrossRef]
9. Wohlin, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In Proceedings of the 18th international conference on evaluation and assessment in software engineering, London, UK, 13–14 May 2014 ; pp. 1–10.
10. Okoli, C.; Schabram, K. A Guide to Conducting a Systematic Literature Review of Information Systems Research. 2010. Available online: https://ssrn.com/abstract=1954824 (accessed on 15 April 2026).
11. Alhirabi, N.; Rana, O.; Perera, C. Security and privacy requirements for the internet of things: A survey. *Acm Trans. Internet Things* **2021**, *2*, 1–37. [CrossRef]
12. Microsoft. Microsoft Threat Modeling Tool. 2020. Available online: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool (accessed on 15 April 2026).
13. OWASP. OWASP Threat Dragon. 2020. Available online: https://owasp.org/www-project-threat-dragon/ (accessed on 3 October 2023).
14. Benton, M. The Knowledge Illusion: Why We Never Think Alone. *Softw. Qual. Prof.* **2017**, *19*, 45.
15. Kohda, Y. Can humans learn from AI? A fundamental question in knowledge science in the AI era. In *Proceedings of the Advances in the Human Side of Service Engineering: Proceedings of the AHFE 2020 Virtual Conference on The Human Side of Service Engineering, USA, 16–20 July 2020*; Springer: Cham, Switzerland, 2020, pp. 244–250.

16. Ziegler, E.; Edelsbrunner, P.A.; Stern, E. The relative merits of explicit and implicit learning of contrasted algebra principles. *Educ. Psychol. Rev.* **2018**, *30*, 531–558. [CrossRef]
17. Ellis, R. 1. Implicit and explicit learning, knowledge and instruction. In *Implicit and Explicit Knowledge in Second Language Learning, Testing and Teaching*; Multilingual Matters: Bristol, UK, 2009; pp. 3–26.
18. Schuchard, J.; Thompson, C.K. Implicit and explicit learning in individuals with agrammatic aphasia. *J. Psycholinguist. Res.* **2014**, *43*, 209–224. [CrossRef]
19. Crowder, J.A.; Carbone, J.; Friess, S. Implicit Learning in Artificial Intelligence. In *Artificial Psychology*; Springer: Cham, Switzerland, 2020; pp. 139–147.
20. Konecki, M.; Kadoić, N.; Piltaver, R. Intelligent Assistant for Helping Students to Learn Programming. In *Proceedings of the 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*; IEEE: Piscataway, NJ, USA, 2015; pp. 924–928.
21. Shekhar, G.; D'Souza, R.; Fernandes, K. *AI-Driven Contextual Virtual Teaching Assistant Using RASA*; Association for Computing Machinery, Inc.: New York, NY, USA, 2020 ; p. 346. [CrossRef]
22. Elragal, A.; Awad, A.I.; Andersson, I.; Nilsson, J. A Conversational AI Bot for Efficient Learning: A Prototypical Design. *IEEE Access* **2024**, *12*, 154877–154887. [CrossRef]
23. Villegas-Ch, W.; Jaramillo-Alcázar, A.; Mera-Navarrete, A. Assistance System for the Teaching of Natural Numbers to Preschool Children with the Use of Artificial Intelligence Algorithms. *Future Internet* **2022**, *14*, 266. [CrossRef]
24. Anderson, F.; Grossman, T.; Fitzmaurice, G. Trigger-action-circuits: Leveraging generative design to enable novices to design and build circuitry. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, Québec City, QC, Canada, 22–25 October 2017; pp. 331–342.
25. Sun, Z.; Anbarasan, M.; Praveen Kumar, D. Design of online intelligent English teaching platform based on artificial intelligence techniques. *Comput. Intell.* **2021**, *37*, 1166–1180. [CrossRef]
26. Pham, X.L.; Pham, T.; Nguyen, Q.M.; Nguyen, T.H.; Cao, T.T.H. Chatbot as an intelligent personal assistant for mobile language learning. In Proceedings of the 2018 2nd International Conference on Education and E-Learning, Bali, Indonesia, 5–7 November 2018; pp. 16–21.
27. Allen, M.; Naeem, U.; Gill, S.S. Q-Module-Bot: A Generative AI-Based Question and Answer Bot for Module Teaching Support. *IEEE Trans. Educ.* **2024**, *67*, 793–802. [CrossRef]
28. Kim, S.G.; Yoon, S.M.; Yang, M.; Choi, J.; Akay, H.; Burnell, E. HeyTeddy: Conversational test-driven development for physical computing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2019**, *3*, 139. [CrossRef]
29. Imtiaz, M.A.; Luxton-Reilly, A.; Plimmer, B. ThinkInk-An Intelligent Sketch Tool for Learning Data Structures. In Proceedings of the Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal, QC, Canada, 21–26 April 2018; pp. 1–6.
30. Drew, D.; Newcomb, J.L.; McGrath, W.; Maksimovic, F.; Mellis, D.; Hartmann, B. The toastboard: Ubiquitous instrumentation and automated checking of breadboarded circuits. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology, Tokyo, Japan, 16–19 October 2016; pp. 677–686.
31. Yuan, X.; Vega, P.; Qadah, Y.; Archer, R.; Yu, H.; Xu, J. Visualization tools for teaching computer security. *ACM Trans. Comput. Educ. (Toce)* **2010**, *9*, 1–28. [CrossRef]
32. Zhao, J.; Wang, H.; Wang, X.; Gou, J. The Research and Implementation of Intelligent Reading Assistant System Framework Based on Knowledge Graph. In Proceedings of the 2021 The 3rd World Symposium on Software Engineering, Xiamen, China, 24–26 September 2021; pp. 45–51.
33. Aldeman, N.L.S.; de Sá Urtiga Aita, K.M.; Machado, V.P.; da Mata Sousa, L.C.D.; Coelho, A.G.B.; da Silva, A.S.; da Silva Mendes, A.P.; de Oliveira Neres, F.J.; do Monte, S.J.H. Smartpathk: A platform for teaching glomerulopathies using machine learning. *BMC Med. Educ.* **2021**, *21*, 1–8. [CrossRef]
34. Peternier, A.; Thalmann, D.; Vexo, F. Mental vision: A computer graphics teaching platform. In *Proceedings of the International Conference on Technologies for E-Learning and Digital Entertainment*; Springer:Berlin/Heidelberg, Germany, 2006; pp. 223–232.
35. Dias, P.; Madeira, J.; Santos, B.S. Using VTK as a Tool for Teaching and Applying Computer Graphics. In Proceedings of the Eurographics (Education Papers), Vienna, Austria, 4–8 September 2006 ; pp. 61–67.
36. Harteveld, C.; Manning, N.; Abu-Arja, F.; Menasce, R.; Thurston, D.; Smith, G.; Sutherland, S.C. Design of playful authoring tools for social and behavioral science. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces Companion*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 157–160.
37. Koch, J.; Lucero, A.; Hegemann, L.; Oulasvirta, A. May AI? Design ideation with cooperative contextual bandits. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019; pp. 1–12.
38. Koch, J.; Taffin, N.; Beaudouin-Lafon, M.; Laine, M.; Lucero, A.; Mackay, W.E. ImageSense: An intelligent collaborative ideation tool to support diverse human-computer partnerships. *Proc. ACM Hum. Comput. Interact.* **2020**, *4*, 1–27. [CrossRef]

39. Boud, D.; Molloy, E. *Feedback in Higher and Professional Education: Understanding It and Doing It Well*; Routledge: Oxfordshire, UK, 2013; p. 2013.
40. Oviatt, S.; Arthur, A.; Cohen, J. Quiet interfaces that help students think. In Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology, Montreux, Switzerland, 15–18 October 2006; pp. 191–200.
41. Hobert, S. Say Hello to 'Coding Tutor'! Design and Evaluation of a Chatbot-based Learning System Supporting Students to Learn to Program. In *Proceedings of the 40th International Conference on Information Systems (ICIS 2019), Munich, Germany, 15–18 December 2019*; Association for Information Systems: Atlanta, GA, USA, 2019; pp. 1–17.
42. Fossati, D.; Di Eugenio, B.; Ohlsson, S.; Brown, C.; Chen, L. Data driven automatic feedback generation in the iList intelligent tutoring system. *Technol. Instr. Cogn. Learn.* **2015**, *10*, 5–26.
43. Michaud, L.N.; McCoy, K.F.; Pennington, C.A. An intelligent tutoring system for deaf learners of written English. In Proceedings of the fourth international ACM conference on Assistive technologies, Arlington, VA, USA, 13–15 November 2000; pp. 92–100.
44. Atilola, O.; Valentine, S.; Kim, H.H.; Turner, D.; McTigue, E.; Hammond, T.; Linsey, J. Mechanix: A natural sketch interface tool for teaching truss analysis and free-body diagrams. *AI EDAM* **2014**, *28*, 169–192. [CrossRef]
45. Obeid, C.; Lahoud, I.; El Khoury, H.; Champin, P.A. Ontology-based recommender system in higher education. In Proceedings of the Companion Proceedings of the The Web Conference 2018, Lyon, France, 23–27 April 2018; pp. 1031–1034.
46. Sermuga Pandian, V.P.; Suleri, S.; Beecks, C.; Jarke, M. MetaMorph: AI Assistance to Transform Lo-Fi Sketches to Higher Fidelities. In Proceedings of the 32nd Australian Conference on Human-Computer Interaction, Sydney, NSW, Australia, 2–4 December 2020; pp. 403–412.
47. Buchanan, S.; Laviola Jr, J.J. Cstutor: A sketch-based tool for visualizing data structures. *Acm Trans. Comput. Educ. (TOCE)* **2014**, *14*, 1–28.
48. Reber, A.S. Implicit learning of artificial grammars. *J. Verbal Learn. Verbal Behav.* **1967**, *6*, 855–863. [CrossRef]
49. Zhao, J.; Bhatt, S.; Thille, C.; Zimmaro, D.; Gattani, N.; Walker, J. Introducing Alexa for E-learning. In Proceedings of the Seventh ACM Conference on Learning @Scale, Virtual, 12–14 August 2020; pp. 427–428.
50. Winkler, R.; Söllner, M.; Neuweiler, M.L.; Conti Rossini, F.; Leimeister, J.M. Alexa, can you help us solve this problem? How conversations with smart personal assistant tutors increase task group outcomes. In Proceedings of the Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019; pp. 1–6.
51. Estevez, J.; Garate, G.; Graña, M. Gentle introduction to artificial intelligence for high-school students using scratch. *IEEE Access* **2019**, *7*, 179027–179036. [CrossRef]
52. Carney, M.; Webster, B.; Alvarado, I.; Phillips, K.; Howell, N.; Griffith, J.; Jongejan, J.; Pitaru, A.; Chen, A. Teachable machine: Approachable Web-based tool for exploring machine learning classification. In Proceedings of the Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April 2020; pp. 1–8.
53. Lin, P.; Van Brummelen, J.; Lukin, G.; Williams, R.; Breazeal, C. Zhorai: Designing a conversational agent for children to explore machine learning concepts. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13381–13388.
54. García, J.D.R.; Moreno-León, J.; Román-González, M.; Robles, G. LearningML: A tool to foster computational thinking skills through practical artificial intelligence projects. *Rev. De Educ. A Distancia (RED)* **2020**, *20, 1–37* .
55. Rodríguez-García, J.D.; Moreno-León, J.; Román-González, M.; Robles, G. Evaluation of an online intervention to teach artificial intelligence with learningml to 10-16-year-old students. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, Virtual, 13–20 March 2021; pp. 177–183.
56. Dizon, G. Using intelligent personal assistants for second language learning: A case study of Alexa. *Tesol J.* **2017**, *8*, 811–830. [CrossRef]
57. Chen, L.; Jing, Q.; Tsang, Y.; Wang, Q.; Liu, R.; Xia, D.; Zhou, Y.; Sun, L. AutoSpark: Supporting Automobile Appearance Design Ideation with Kansei Engineering and Generative AI. In Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology, Pittsburgh, PA, USA, 13–16 October 2024; UIST '24. [CrossRef]
58. Peng, X.; Koch, J.; Mackay, W.E. DesignPrompt: Using Multimodal Interaction for Design Exploration with Generative AI. In Proceedings of the 2024 ACM Designing Interactive Systems Conference, Copenhagen, Denmark, 1–5 July 2024; DIS '24, pp. 804–818. [CrossRef]
59. Gao, J.; Peng, B.; Li, C.; Li, J.; Shayandeh, S.; Liden, L.; Shum, H.Y. Robust Conversational AI with Grounded Text Generation. *arXiv* **2020**, arXiv:2009.03457. [CrossRef]
60. Ong, R.; Raof, R.; Sudin, S.; Choong, K. A Review of Chatbot development for Dynamic Web-based Knowledge Management System (KMS) in Small Scale Agriculture. In *Proceedings of the Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2021; Volume 1755, p. 012051.
61. Boonstra, L. Introduction to Conversational AI. In *The Definitive Guide to Conversational AI with Dialogflow and Google Cloud*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 1–27.

62. Wellnhammer, N.; Dolata, M.; Steigler, S.; Schwabe, G. Studying with the help of digital tutors: Design aspects of conversational agents that influence the learning process. In Proceedings of the 53rd Hawaii International Conference on System Sciences, Maui, HI, USA, 7–10 January 2020; pp. 146–156.
63. Kim, H.; Lim, Y.K. Teaching-learning interaction: A new concept for interaction design to support reflective user agency in intelligent systems. In Proceedings of the Designing Interactive Systems Conference 2021, Virtual, 28 June–2 July 2021; pp. 1544–1553.
64. Reilly, E.D. Interactive system. In *Encyclopedia of Computer Science*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 907–908.
65. Major, N.; Ainsworth, S.; Wood, D. REDEEM: Exploiting symbiosis between psychology and authoring environments. *Int. J. Artif. Intell. Educ.* **1997**, *8*, 317–340.
66. Mitrovic, A. Modeling domains and students with constraint-based modeling. In *Advances in Intelligent Tutoring Systems*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 63–80.
67. Herbert, B.; Billinghurst, M.; Weerasinghe, A.; Ens, B.; Wigley, G. A generalized, rapid authoring tool for intelligent tutoring systems. In Proceedings of the 30th Australian Conference on Computer-Human Interaction, Melbourne, Australia, 4–7 December 2018; pp. 368–373.
68. del Vado Vírseda, R.; Fernández, P.; Muñoz, S.; Murillo, A. An Intelligent Tutoring System for Interactive Learning of Data Structures. In Proceedings of the Computational Science–ICCS 2009: 9th International Conference Baton Rouge, LA, USA, 25–27 May 2009; pp. 53–62.
69. Khine, M.S. AI in teaching and learning and intelligent tutoring systems. In *Artificial Intelligence in Education: A Machine-Generated Literature Overview*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 467–570.
70. Aamir, M.; Bhusry, M. Recommendation system: State of the art approach. *Int. J. Comput. Appl.* **2015**, *120*. [CrossRef]
71. Stolterman, E.; Pierce, J. Design tools in practice: Studying the designer-tool relationship in interaction design. In Proceedings of the Designing Interactive Systems Conference, Tyne, UK, 11–15 June 2012; pp. 25–28.
72. Diefenbach, D.; Lopez, V.; Singh, K.; Maret, P. Core techniques of question answering systems over knowledge bases: A survey. *Knowl. Inf. Syst.* **2018**, *55*, 529–569. [CrossRef]
73. Lan, Y.; He, G.; Jiang, J.; Jiang, J.; Zhao, W.X.; Wen, J.R. Complex knowledge base question answering: A survey. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 11196–11215. [CrossRef]
74. Hill, R.K. What an algorithm is. *Philos. Technol.* **2016**, *29*, 35–59. [CrossRef]
75. Burkett, I. An introduction to co-design. *Sydney Knode* **2012**, *12*, 12.
76. Terzopoulos, G.; Satratzemi, M. Voice assistants and artificial intelligence in education. In Proceedings of the 9th Balkan Conference on Informatics, Sofia, Bulgaria, 26–28 September 2019; pp. 1–6.
77. Seo, H.; Hwang, T.; Jung, J.; Kang, H.; Namgoong, H.; Lee, Y.; Jung, S. Large language models as evaluators in education: Verification of feedback consistency and accuracy. *Appl. Sci.* **2025**, *15*, 671. [CrossRef]
78. Izacard, G.; Lewis, P.; Lomeli, M.; Hosseini, L.; Petroni, F.; Schick, T.; Dwivedi-Yu, J.; Joulin, A.; Riedel, S.; Grave, E. Few-shot learning with retrieval augmented language models. *arXiv* **2022**, arXiv:2208.03299. [CrossRef]

https://doi.org/10.3390/ai7050160