Intelligent Design Techniques Towards Implicit and Explicit Learning: A Systematic Review

Fatmah Alqarni
School of Computer Science and Informatics
Cardiff University, UK
Alqarnif@cardiff.ac.uk

Omer Rana
School of Computer Science and Informatics
Cardiff University, UK
Ranaof@cardiff.ac.uk

Charith Perera
School of Computer Science and Informatics
Cardiff University, UK
pererac@cardiff.ac.uk

November 3, 2023

Abstract

Artificial intelligence techniques are advancing in areas like health, cooking, education, and agriculture, aiming to educate users and enhance their knowledge. This survey reviews design tools and their features, introducing a classification of implicit and explicit features from user interactions. An analysis of 35 studies reveals the application of these techniques, detailing knowledge storage, interaction types, algorithms, and evaluation methods. Key findings include the potential of techniques for both implicit and explicit learning, a lack of detailed information in some implementations, and a highlighted need for tools that educate novice software engineers about privacy through design IoT applications.

***K*eywords** AI tool, Explicit learning, Implicit Learning, AI Conversational tool, AI mediated in teaching

1 Introduction

Learning is an important process for acquiring knowledge, developing skills, and making decisions. According to Sequeira [1], learning involves developing new skills, changing perspectives, and understanding concepts. For example, Riivari et al. [2] highlighted that learning teamwork skills through Novicraft, which is a computer game, improving students' skills, such as identifying their team roles and how to effectively make decisions in the team. Artificial intelligence (AI) has been instrumental in guiding and helping users learn through activities and virtual systems. AI techniques have shown remarkable progress and their potential applications are increasing in many fields, such as health, cooking, education, and agriculture. Furthermore, a significant improvement in the overall performance of 1500 companies was observed when humans and AI worked together [3]. AI assistance for design tools aid designers in working faster with improved performance and efficiency. Thus, improving companies' competitiveness in a rapidly changing market [3]. In the sphere of privacy and security, it has been stated that identifying threats early in the design phase can reduce the cost and time that may take later in the development process [4]. To facilitate this, various tools have been developed to identify security and privacy threats. Microsoft threat modelling tool uses the STRIDE methodology to identify security threats [5], while the LINDDUN methodology is used for privacy [6]. Additionally, the Threat Dragon tool can be used to identify both security and privacy issues [7].

AI assistant techniques are implemented as virtual systems, design systems, teaching platforms, and conversational tools to provide specialised assistance to users. As Baylor and Kim [8] noted, the role of Pedagogical agents for academy students in the learning process is of particular importance, as they can augment knowledge acquisition, enrich learning and motivation, and enhance self-confidence. Notably, such agents can have an invaluable impact in the educational field, potentially encouraging students to learn and achieve more.

The following research questions (RQs) will be answered through this review:

- RQ1: What interactions of humans with AI assistant techniques have been documented in the literature?
- RQ2: What types of learning do users acquire while using AI assistant techniques?
- RQ3: How AI assistant techniques have been implemented in the literature? (i) How knowledge is stored and continuously updated?, (ii) Who are the stakeholders interacting with the system?, (iii) What algorithm and techniques is used?, (iv) How do they evaluate?, (v) What types of output are generated?

The contributions of this paper are as follows:

- A comprehensive review design tools and characteristics (i.e., security and privacy) and design prototyping tools that support teaching and learning.
- Propose a taxonomy to effectively classify the types of learning in AI techniques.
- Review the AI techniques implementation to gain an understanding of how these techniques represent Knowledge and make decisions.

The remainder of this paper is organised as follows: Section 2 presents a methodology used to select papers. Next, Section 3 provides an overview of design tools and their respective characteristics; Section 4 focuses on implicit and explicit learning; while Section 5 discusses the approach to implementation of some techniques, and then AI techniques for implementing intelligent behaviours are discussed in Section 6 and then highlight the research challenges and directions in Section 7. Finally, Section 8 concludes the literature review.

2 METHODOLOGY

In order to retrieve sufficient and high-quality papers, we follow the guidelines and snowballing procedure outlined in the systematic literature in [9]. 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). 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 backforward snowballing to find more relevant articles. As Wohlin [9] mentioned, Forward snowballing is conducted by looking at the titles of papers that cite the source, while Backward snowballing is undertaken by examining the titles of papers in papers in the source's reference list.

In each phase of the search process, reading the abstract and conclusion of a paper was a step to making a decision, either excluding or extracting the paper. All relevant papers were in a period from 1960 to 2022. Papers that did not provide teaching and helping for users through AI were excluded; furthermore, since the scope of AI is wide, robotics was excluded since it falls outside the boundaries of this particular search. Consequently, all papers included in the Literature Review corresponded to the research questions.

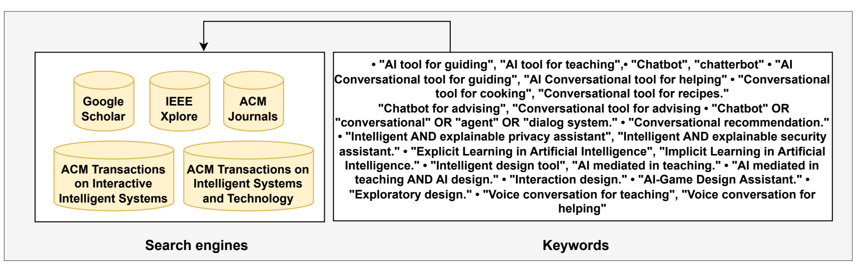


Figure 1: Keywords and search engines used for search query

3 Overview of Design Tools and Characteristics

Incorporating privacy requirements in the design and implementation of systems are challenging and necessitating for the translation of complex social, legal and ethical issues into systems requirements [10]. To aid in these efforts, the notion of "privacy by design" has been proposed as a guide principle [10]. Privacy by Design (PbD) is a proactive approach that embeds privacy early in the design phase, and this makes occurrences of privacy threats excluded in products [11]. Section 3.1 discusses some privacy and security design tools that emphasize privacy and security by design, while Section 3.2 presents design tools that can be used for teaching and learning.

3.1 Privacy and Security Design Tools

It has been stated that identifying threats and software vulnerabilities in a design phase could be reduced the overwhelming majority of risks and cost in the following steps in the systems development life cycle (SDLC) [4]. Microsoft is one of the most widely contributed to the development of SDLC. It profferes a Microsoft threat modelling tool which aims to create more secure software [5]. Microsoft threat modelling uses the STRIDE model, which is an abbreviation for six categories, Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege [5]. The tool follows a process for mitigating threats, as illustrated in Figure 3.

Firstly, create a Data Flow Diagram (DFD) using a drag-and-drop feature to easily draw the diagram scenario. The term of DFD means a visual notation that concentrates on the basic, functional parts of early software designs [12]. DFD in Microsoft threat modelling includes Data flows, Data stores, Processes and Interactors. Identifying and mitigating threats is the next process which is used the STRIDE model to identify threats. The final step is validation, allowing users to mitigate these threats. Table 1 shows that each DFD is mapped to potential security threats, marked with a star symbol to indicate their severity.

Table 1: Mapping DFD with STRIDE model

	DFD elements			
STRIDE	Data flows	Data stores	Processses	Interactors
Spoofing			*	*
Tampering	*	*	*	
Repudiation			*	*
Information disclosure	*	*	*	
Denial of service	*	*	*	
Elevation of privilege			*	

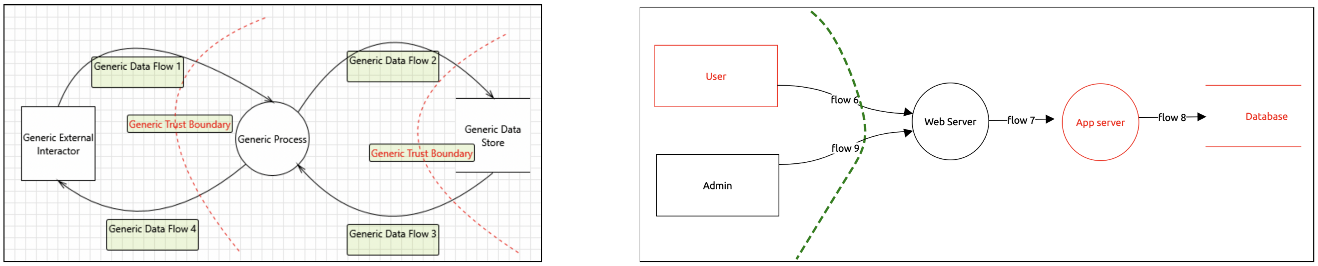
Threat dragon is another method for threat modelling supported by the Open Web Application Security Project (OWASP) [13]. The tool can support STRIDE types, and CIA [13], as well as the recently added LINDDUN. Threat dragon is similar to Microsoft threat modelling in that both tools are open source and support STRIDE. However, Threat dragon is cross-platform and can run on Windows, Mac, and Linux [7], whereas Microsoft threat modelling is limited to Windows [14]. Threat Dragon is limited in terms of the number of stencils available, in comparison to the wide range of stencils provided by Microsoft Threat Modeling. Furthermore, Microsoft Threat Modeling automatically generates potential threats with a detailed description of each, while Threat Dragon requires manual creation of each node in the Data Flow Diagram (DFD) and the subsequent traversal of each node to identify potential threats, as illustrated in Table 2 which compares these two tools.

Table 2: Microsoft threat tool vs. OWASP Threat Dragon

	Microsoft threat modelling tool	**OWASP threat dragon**
Covered Scope	Security	Combined security and privacy
Support	STRIDE	STRIDE, CIA and Privacy
Platform	Only one platform	Cross-platform
Stencils	Various	Limited
Manage threats	Entirely	partially
Documentation	Comprehensive all threats with details	Per each element with a few details

An example of how the Microsoft Threat Modeling Tool and Threat Dragon utilize Data Flow Diagrams is demonstrated in Figure 2. As [14] points out, each of the elements in the DFDs mentioned in the diagrams is vulnerable to a range of potential threats.

Another Framework inspired by STRIDE is LINDDUN, a methodology that applies privacy threat modelling. LINDDUN is an acronym for Likability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness and Non-compliance [6]. Figure 3 illustrates the approach that LINDDUN follows to manage and mitigate threats. The first step is to allow users to create the scenario of a DFD and manually elicit threats by mapping DFD elements to threat categories, as explained in Table 3. This repetition of DFD elements helps to identify potential privacy issues.



(A) Microsoft threat modelling tool

(B) Threat dragon

Figure 2: In two different models, Data Flow Diagrams (DFD) are employed as a visual representation of system processes. Within these DFDs, security threats are depicted conspicuously in red, emphasizing their critical role in system vulnerability assessments and risk mitigation.

Table 3: Mapping LINDDUN model with DFD

	DFD elements			
LINDDUN	Data flows	Data stores	Processses	Interactors
Linkability	*	*	*	*
Identifiability	*	*	*	*
Non-repudiation	*	*	*	
Detectability	*	*	*	
Disclosure of information	*	*	*	
Unawareness				*
Non-compliance	*	*	*	

Finally, appropriate solutions are identified to address the exposed threats. As mentioned in Table 3, each DFD element is susceptible to a set of LINDDUN threats, and their intersections are marked with a star symbol which indicates a potentiate privacy threat.

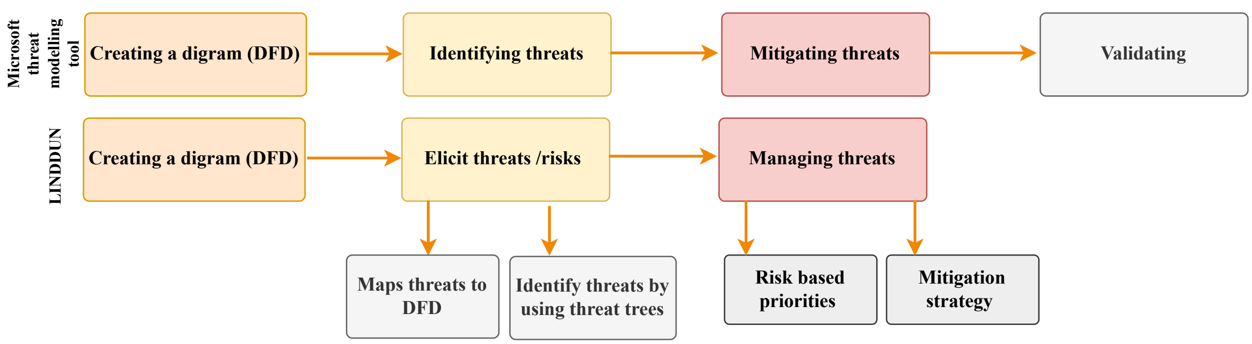


Figure 3: The methodology of Microsoft threat modeling and LINDDUN.

3.2 DESIGN PROTOTYPING TOOLS

Popular collaborative design tools help designers create and collaborate with teams or peers. Scratch is a block-based programming tool created to help children aged 8 to 16 learn how to programme when they work on a project, such as games or short stories. The design purpose of Scratch is vital to encourage the idea of autodidacticism learning through starting to think and cooperate with peers [15]. Furthermore, Scratch has been used as a teaching tool for second-language teaching [16], and AI for High-School Students [17]. Shaath et al. [18] developed an intelligent tutoring system to assist students who want to learn Photoshop. The learning process is based on material, ie . lessons, questions, and exercises. This allows students to capture their lectures and follow up whenever they want.

4 Explicit and Implicit Learning Features

As discussed in Section 1, learning is the first step for humans to acquire knowledge and achieve goals in any domain. Benton [19] mentioned that humans rely on more than just the knowledge stored in their brains but also need knowledge from different sources such as the environment and other people. To facilitate such learning, humans need efficient tools or systems to educate them on what they need to know to improve their understanding. Kohda [20] infer that humans achieve more with support from AI. Therefore, humans can learn explicitly or implicitly through their interaction with AI (see Figure 4). Furthermore, a taxonomy of learning features is provided to aid in understanding the different learning types . Sections 4.1 and 4.2 provide more details for each type of learning and illustrate how AI can be leveraged to augment them.

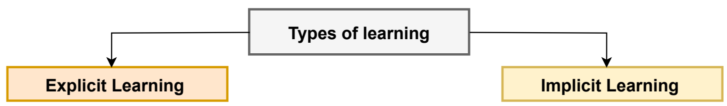


Figure 4: Types of learning in AI techniques.

4.1 Explicit Learning

Ziegler et al. [21] defined explicit learning as a type of learning in which learners are directly instructed or taught for a particular purpose. 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 realize their aims and objectives to achieve their goals [21]. As such, Ellis [22] mentioned 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.

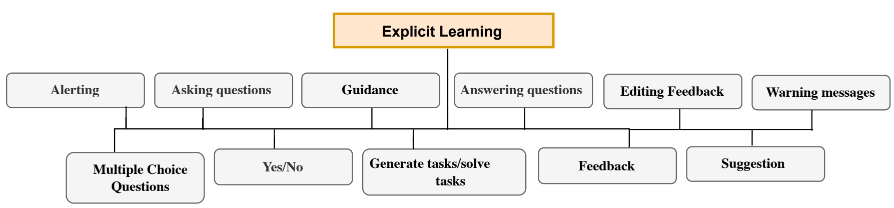


Figure 5: Types of explicit learning through interactions users with AI techniques.

4.1.1 Asking Questions

In order to assist learners, asking questions could be a useful way to check learners' understanding as well as encourage them to be active with the system. In the health field, patients learn about their disease through interaction with an AI doctor system [23]. The AI doctor asks questions about patients' symptoms; based on the answer to these questions, the AI doctor can infer a proper diagnosis. Moreover, AI doctor can increase patients' awareness about the types of infections that cause diseases. For example, by typing the query asking about fever, the system lists names caused by fever, such as bacteria, viruses, etc [23].

In the programming, students have asked an intelligent assistant question to learn concepts in programming (i.e. the meaning of programming or array), and the intelligent assistant replies to the related question[24]. In this study, they found the intelligent assistant increased students' motivation toward learning programming and made learning more effortless and more enjoyable. For learning cooking, a user can ask the tool for new recipes, create a recipe from a few ingredients in the fridge, or request suggestions recipes on how to reduce food waste [25]. In education, contextual help provides instructor resources in the form of a Contextual Virtual Teaching Assistant (CVTA) [26]. CVTA can ask questions and provide tailored answers based on Tutors' materials. CVTA can also help the Instructor identify students' needs.

4.1.2 Yes or No Questions

Some systems designed the interaction via closed questions such as [27]. These questions are easy for answers and may help users make decisions quickly. For example, when users ask about adding an item to a meal, they may want to know if it is compulsory or not. Villegas-Ch et al. [28] developed a system to assist preschool children in identifying, replicating, and writing integer numbers from zero to nine. The dialog between the child and the system was based on questions and answers, most of which required is designed by yes or no voice responses. It was observed that the motor skills in writing were enhanced for children who used the system.

4.1.3 Multiple choice Questions

Users can learn from the explicit options listed on the screen, helping them to consider and select the most appropriate answer. The Trigger-Action-Circuits system [29] aids novices in designing circuits by generating multiple circuit options on the screen, and a user can choose suited to their use case based on different needs, such as cost and component availability. The result showed that users who used TAC could complete tasks assigned to them more successfully and quicker than the other group who did not use TAC. In the DL-OIET system [30], users answer questions by selecting from multiple-choice options. Users have reported positive feedback after using the DL-OIET system, with many noting an improvement in their English language skills.

4.1.4 Answering Questions

Conversation tools can answer farmers' questions in [31, 32, 33]. Farmers can learn bout horticulture, government schemes, market costs, farming and weather in [31]. Farmers can ask questions and the chatbot obtain answers [33]. Likewise, literate farmers in India learn from answers provided by FarmChat when they ask questions related to farming [32]. Therefore, agriculturalists learn how to make the right decisions about their products by learning from these tools. English Practice is a tool designed to help international learners who want to learn the English language. Learners use materials in the tool, i.e. (quizzes, vocabulary or grammar lessons) to answer questions provided by the tool [34].

4.1.5 Guidance

Several researchers have suggested systems to support novices on their hardware journey. HeyTeddy assists novice programmers in learning programming Language by using Arduino [35]. It guides users step-by-step with instructions on how to build the circuit. For example, when users ask HeyTeddy how to use a motor, HeyTeddy guides the user until completing the task. As a result, students feel more confident in building circuits with HeyTeddy. Thus, HeyTeddy was useful for users to avoid mistakes while using Arduino. Smart Care [36] is designed to guide pregnant women in rural areas in Bangladesh and aims to increase their awareness during pregnancy. This intelligent assistant advises on the health and symptoms that may occur each month.

4.1.6 Warning Message

Khilji et al. [27] used the questions and answers method in the cooking system; One type of response to users explicitly in the form of a warning. Furthermore, ThinkInk [37] provides useful warning messages when errors are made in the task, allowing users to identify and fix any mistakes quickly. With their interactive learning environments and helpful feedback, this effective tool for beginner programmers to improve their data structure skills and knowledge.

4.1.7 Generate Tasks or Solve Tasks

Users perform some tasks or complete activities that may encourage them to understand better. For example, the Toastboard system in [38] allows users to debug circuits easily and helps them learn how to assemble and validate simple circuits. Likewise, an interactive system introduced a tutorial with interactive tests to help a user writes a program for Arduino or assembles a circuit following predetermined steps [38]. Therefore, users understand the content and avoid errors.

Previous studies conducted by [39] to develop animated visualization tools for teaching Computer Security. Tools covered were dedicated to some concepts, i.e.(the visualization tool for wireless network attacks). Students can engage in learning from these tools by controlling the animation process, such as by clicking on (stop, pause, and resume). Also, students can learn from the description message that is appeared on the user interface to provide a dynamic update description of the case animation. Students who used these tools provided positive feedback. The tools were easy and helpful in understanding information about security concepts.

Zhao et al. [40] provided the system that helps educators save the time they spend preparing teaching materials and making test questions. Also, the system is designed with the aim of assisting learners by providing reading comprehension assistance tasks through knowledge graphs. Students should solve these tasks that are included test questions associated with the text.

Aldeman et al. [41] developed a platform for teaching glomerulopathies to students and pathology instructors. Students learn through materials and solve activities, i.e. quizzes and image analysis.

Mental Vision [42] consists of several interactive applications called modules. A teacher can share one model with students in the classroom, e.g. a Bezier surface, and allow them access remotely. Students can play to modify the parameters visualisation and display in the board classroom. Students' motivation was positive and improved because they could develop complex visualisations in a brief time. The VTK [43] helps students gain knowledge about visualisation by allowing students to develop tasks. Students found using the tool helpful in understanding and applying computer graphics concepts and creating successful projects.

Users can create and learn how to design experiments using the tools available [44]. Through these tools, users can create playable experimental scenarios by making modifications to text, symbols, and other elements. However, the authors emphasise the need for artificial intelligence assistance to ensure users can successfully achieve a basic level of experiment design, even with prior instructions.

4.1.8 Suggestions

Recipe Bot in [45] provides a combination of question-answering and recommendations. It helps users find recipes when they ask specific questions about dishes; it returns recipes of these dishes or suggests recipes based on answering some questions when users do not have any idea what they want to eat. In the same way, a suggestion tool was built by Angara et al. [46] is also recommended recipes to users according to their taste preferences and health situation. Users have reported that using the tool is effective in reducing food waste and managing grocery budgets, as well as providing personalized recipe recommendations that take into account personal contexts such as food allergies and dietary goals [46].

Patients ask the "HealthAssistantBot" chatbot questions about their health issues, and AI provides suggestions about their disease and may suggest the nearest dr [47]. Users who interacted with HealthAssistantBot in regular use scenarios gave favourable feedback, showing that it is effective in helping patients take care of their health. The chatbot [34] suggests hints or helps to explain concepts more. Guidance may help learners to learn the English language quickly and at any time. Although this tool was generally helpful in learning, the English Practice could not respond when users used voice.

A study conducted Koch et al. [48] found that the use of May AI assistance in the design phase helped designers quickly find relevant ideas and inspiration for their projects. The AI assistant provided designers with access to visually relevant materials such as pictures. Similarly,ImageSense [49] help a designer collaborate with another remote designer with allowing suggestions from AI. Both May AI and ImageSense help designers to find creative ideas and inspiration faster.

4.1.9 Alerting

Another a way for AI systems to notify users of potential issues or discrepancies that may require attention. Alerts are usually designed to be triggered by certain conditions, such as alerting students about upcoming deadlines [26], reminding users to take their medication [47], or sending notifications for mandatory checkups with a doctor [36].

4.1.10 Feedback

According to [50], feedback serves as a mechanism for learners to determine their level of success in their work and whether they are meeting expectations. Oviatt et al. [51] mentioned that digital tools could provide real-time feedback for users, which is much more efficient than pen-and-paper methods. An assistant tool called 'coding tutor' in the reference [52] has provided explanations to students who ask about coding exercises or answer open questions and provides Feedback on assignments provided by students. This coding tutor was helpful for novice programmers who needed to understand algorithms or apply code.

Microsoft threat modelling [5] provides feedback to a user who designs the system and wants to learn about the potential threats in the system before moving to the development phase. The tool lists all potential risks with descriptions and classifies them by different colours to make the user able to understand each threat; the red colour appears when the risk is high, orange means the risk is medium and low-priority threats are labelled as grey.

Students request analysis for a piece of writing in this study [53]. The system determines grammatical errors in the text by highlighting sentences with grammar errors and presenting the text to the students as feedback. In addition, students can correct grammar errors and resubmit their writing again. Therefore, the system improves the literacy of the deaf and makes them learn without feeling the shame of mistakes they might feel with a human teacher.

Atilola et al. [54] provides explicit feedback to students, informing them if their problem-solving steps are correct or incorrect. The feature of using this tool does not provide the answers to the students, but it informs them of any mistakes they have made and what those errors are. Through the use of Mechanix, students can acquire a deeper understanding of truss analysis and free-body diagrams. This will enable them to produce accurate diagrams faster and with increased assurance in their results. Obeid et al. [55] provides students with feedback to help them make informed decisions when selecting an academic institution and major that aligns with their interests. Sermuga Pandian et al. [56] contributed to the design process by transforming low-level sketches into high-quality sketches, thereby streamlining the workflow and saving designers time and effort compared to manually reworking the sketch. Fossati et al. [57] developed iList, which provides five types of feedback to users: syntax error, execution feedback, final feedback after submitting a problem, and both reactive and proactive feedback. This fifth version of iList proved to be beneficial in guiding students towards successful solution paths.

4.1.11 Editing Feedback

Some systems are designed to provide feedback and enable users to make necessary modifications based on this feedback. The Microsoft threat modelling tool [5] offers users the capability to make decisions about feedback by allowing them to either accept or modify the information generated by the tool. Similarly, users can get feedback and modify threats in the Threat Dragon tool [7]. These features help users to recognize their errors and identify potential threats that could affect their systems. Furthermore, it assists them in editing and improving their ability to think and decide, as well as to remember.

ThinkInk [37], and CS Tutor [58] offer interactive environments to facilitate novice programmers in learning data structures with different learning features. ThinkInk provides a visually intuitive representation of the data structure, allowing users to quickly and easily grasp its components. Additionally, CS Tutor allows users to create diagrams of the data structure and have code automatically generated from it. 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 to understand the knowledge.

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 [21]. This mean, It happens without learners noticing or intentionally trying to learn [59]. Reber [60] was the first to explore this concept. His study covered how learners can understand to answer structural relations without using strategies or being able to express the knowledge in words. Crowder et al. [61] have explained this term in which learners do not realise that they are involved in the approaches or activities they undertake.

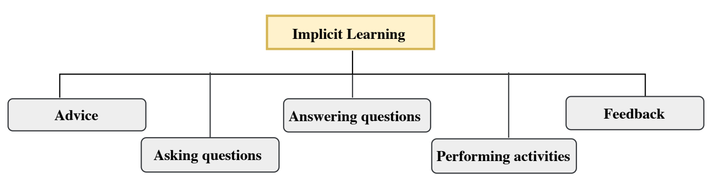


Figure 6: Types of Implicit learning through interactions users with AI techniques

4.2.1 Advice

Implicit advice can help humans by providing subtle prompts that encourage them to reflect on their behaviour or gain new knowledge. These nudges may help people to make positive changes in their lives and develop new skills and insights. Elderly people can tap one choice on the screen or answer using their voice in interaction with the assistant tool [62]. The assistant tool provides helpful advice to the related question in order to keep them more active and healthy. Hence, trying to help the elderly adhere to the advice provided by the tool could implicitly increase their ability to change their behaviour to be healthier.

4.2.2 Asking and Answering Questions

Helping learners develop skills is another way of implicit learning through asking and answering questions way. As Zhao et al. [63] explained the role of Alexa in e-learning to help learners develop skills such as interviewing skills. Alexa can play the role of a peer or adviser based on the scenario. A user can ask the first question, and if Alexa expects the answer to these questions, it continues to follow up on questions from the user; otherwise, Alexa changes the role to an adviser if the user response does not expect by asking some questions and providing guidance. Therefore, users can learn how to improve their skills while asking or answering questions.

Winkler et al. [64] used Alexa to aid a group of users in problem-solving by thinking and asking questions related to the problem. They found the benefit of using Alexa when compared with another group's interaction with human tutors. It was clear the group who used voice assistance understood the problem, and their contribution was homogeneous towards solving the problem. Hence, the group acquire some skills such as collaboration and problem-solving skills. Diabetic patients can learn about their symptoms and how to lead a balanced life between prevention and their desired lifestyle by asking questions [65]. They can ask questions related to the list presented by the system and receive answers from the tool.

4.2.3 Performing Activities

Engaging in activities could be a form of implicit learning that encourages users to complete specific tasks in order to gain a better understanding of concepts. In the AI domain, students were able to solve exercises within the workshop setting by using Scratch [17]. This tool allowed them to gain an understanding of the underlying computational thinking behind AI processes and explore the behaviour of algorithms.

Researchers have contributed to promoting children to understand AI at an early age. Beginner learners learn the concept of machine learning (ML) classification without the required background knowledge in ML coding [66]. This is usedful for learners and instructors alike, as it allows for the introduction of ML to students in an accessible and intuitive way. In the same Vein, Zhorai [67] educate children about ML and knowledge representation. For example, children teach and train Zhorai some facts and Information about animals; and then ask Zhorai some questions to understand what the capability of Zhorai to understand and what it misses. The LearningML platform [68, 69] also assists non-technical in learning ML. Users need to add datasets to feed LearningML and train the model. Finally, they can feed it new data to evaluate LearningML. So, this activity provides users insight into the process of ML and also encourages them to think and learn about ML.

4.2.4 Feedback

Implicit feedback can be a valuable asset in the learning process. It provides learners with immediate and direct feedback on their performance, allowing them to adjust their strategies and improve their understanding of the material. Alexa provides implicit feedback to learners by providing them with an opportunity to practice their pronunciation and language skills in a conversational setting [70]. Also, by providing immediate feedback on pronunciation accuracy, Alexa can help learners improve their speaking ability in a shorter amount of time than traditional methods. The Figure 6 illustrates the different types of implicit learning that can be employed to educate learners without any deliberate intent to learn. Table 4 provides a glimpse into the explicit and implicit learning techniques used in AI assistants and the domain to which they belong.

5 Implementation Approaches

This section discusses the implementation approaches used to develop technique. To gain further insight into the implementation process, we analyze 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 5.1,5.2,5.3 and 5.4 discuss each of these approaches in details.

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 [46, 65, 67], "task-oriented bots", "task bots" in short [71], "bot" in [47] or in some research called "chatbots" [25, 52, 62, 72]. A chatbot is defined as a user interface that is created to mimic chats with users online, typically through text or voice interactions [73]. Chatbots are designed to automate conversations and provide more efficient, accurate, and personalized services to users.

5.1.1 Rule-Based Chatbots

The old-school way of building rule-based chatbots was inflexible because chatbots are designed with specific instructions by following pre-specific rules to address specific inquiries rather than understanding the conversation context [46].

Table 4: Summaries Explicit and Implicit Learning in AI Techniques Across Various Domains, Presented in a Grayscale to Bluescale Spectrum: Commencing with **Health**, **Programming**, **Cooking**, **Education**, **Microsoft** (in Grayscale), Progressing to **Cognitive**, **Skills**, **Agriculture**, **AI**, and **Computer Science** (in Bluescale).

Ref.	**Learning type**		**Brief description of learning types**
	Explicit	Implicit	
[23]	✓		AI asks questions/Patients answer questions
[62]		✓	Advice for senior
[47]	✓		Ask questions/ Providing suggestions/ Medication alert
[65]		✓	Users ask questions/Answer questions
[24]	✓		Ask questions/Answering questions
[27]	✓		Ask questions/Answering questions/Y/N questions/Advising users/ Warning messages
[45],[46]	✓		Suggest recipes to users
[25]	✓		Users ask questions/Answer questions
[26]	✓		Students ask questions/Guidance students by providing answers/Alerting for deadlines
[52]	✓		Provide feedback/Suggestions to students
[28]	✓		Yes/No questions
[54]	✓		Feedback to engineering students.
[53]	✓		Feedback to users /Allow users to edit it
[30]	✓		Ask questions/Answer Questions
[39]	✓		Controlling the animation process.
[42],[43]	✓		Modify an parameters/Develop tasks
[41]	✓		Performing activities
[34]	✓		Answer questions/Hint suggestion
[37]	✓		Providing feedback /Performing activities/Warning messages
[58],[57]	✓		Providing feedback
[5]	✓		Feedback to users /Allow users to edit it
[29]	✓		Answering questions
[38]	✓		Performing tasks
[40]	✓		Generate tasks/Solve tasks
[35]	✓		Advise users/ provide feedback/Alert users
[63],[64]		✓	Ask questions/Answering questions
[44]	✓		Performing activities
[31],[32]	✓		Answering questions
[66],[67]		✓	Performing activities
[69]		✓	Performing activities
[17],[68]		✓	Performing activities
[48],[49]	✓		Providing suggestions
[36]	✓		Gauidence pregnant women/Appointments alert
[56]	✓		Providing feedback

5.1.2 AI Chatbots

In contrast to the chatbot based on rules, an AI chatbot can understand the context by learning from information gathered; Ong et al. explained this in [72]; it can be called a "conversational agent" or "Machine learning Chatbot", which is relay 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. Wellnhammer et al. [74] mentioned that it is better to train systems based on the rule with massive keywords to ensure the output to users is accurate, 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. Chu [45] has combined the rule-based chatbots and AI chatbots approaches). Table 5 summarises the differences between rule-based chatbots and AI chatbots. Hobert [52] combined a "chatbot-based system" built with natural language with an "intelligent programming tutor" (a type of automatic assessment system) could be helpful to provide the proper learning support. This technique could help novice programmers by answering their questions and guiding them through programming tasks. Pham et al. [34] implemented the chatbot, English Practice, on mobile devices and interacted with learners through a window chat.

Miura et al. [62] implemented a virtual caregiver system using a mobile chatbot to enhance the personalised health care of elderly people. A text and speech-based chatbot in [65] to guide patients' behaviour. Polignano et al. [47] built a chatbot by Telegram to provide a personalized health service for users, which is a popular messaging platform; the reason they chose this platform is that Telegram widely used and accessible, and also they did not need to start from scratch to build the bot.

Table 5: 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 conversational kitchen assistant for providing users with recipes for cooking; Angara et al. [46] built a chatbot with IBM Watson technology, and the chatbot developed by Amato and Cozzolino [25] was exploit semantic and natural language processing techniques. A chatbot is designed for helping farmers such as Momaya et al. [31] used RASA X to implement Farmer Chatbot. Rasa X is an open-source tool based on machine learning for automated conversations based on text or sound [31]. Jain et al. [32] used IBM's Watson Conversation service to identify the intent and entity of the user's request. Shekhar et al. [26] implemented the contextual virtual teaching Assistant (CVTA), which requires the utilization of RASA, an open-source conversational AI platform. This AI chatbot is used to construct a virtual assistant that is able to communicate with pupils and give them contextual support.

5.2 AI Systems

Intelligent systems have seamlessly woven themselves into our daily lives, becoming an integral part of our routines [75]. 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.

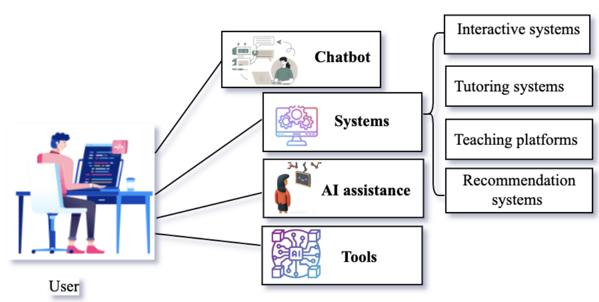


Figure 7: Types of Implementation Approaches and User Interactions

5.2.1 An Interactive System

An interactive system refers to a computer system where users can engage with a running program by providing it with data or instructions using input devices like a keyboard or a mouse [76]. Anderson et al. [29] built an interactive system," Trigger-Action-Circuits (TAC)", to help users, even those with no prior electronics knowledge, to design and build circuits. A Prolog expert system (APES) is used to implement AI conversation between patients and the system [23].

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 [77],[78] and [79]. Koedinger et al. [80] implemented PAT, which is an intelligent tutoring system created to help students learn and understand algebra. Michaud et al. [53] used tutoring systems to aid learners. Michaud et al. [53] developed an intelligent tutoring system to facilitate deaf students in improving their English literacy skills, particularly writing. Another intelligent tutoring system is provided by [57] to provide automated feedback students for learning data analysis. del Vado Vírseda et al. [81] developed an Intelligent tutoring system to teach data structure, while [57] built an especially tailored system for teaching linked lists.

5.2.3 Teaching Platform

AI teaching systems serve as a platform for learners to take charge of their own learning, providing them with a wide array of resources and tools. Furthermore, they gather vast amounts of data on how students learn, offering valuable and continuously evolving dynamic resources that enhance online education [82]. In Pedagogical visualization, an Intelligent platform is implemented by using machine Learning for teaching glomerulopathies [41]. Sun et al. [30] built an online English teaching platform that utilizes artificial intelligence and knowledge recommendation to create a modern tool for students.

5.2.4 Recommendation System

A Recommender System is a combination of software tools and machine learning methods designed to offer valuable suggestions for items or services based on a user's preferences [83]. An intelligent reading system is implemented by Zhao et al. [40] to apply cognitive intelligence to reading comprehension teaching. Khilji et al. [27] used recommendations algorithms for recommended recipes. Obeid et al. [55] implemented a recommender system to identify students' interests and skills, and provide personalized advice based on these.

5.3 AI assistance

Konecki et al. [24] proposed an intelligent assistant for helping students who are interested in learning Programming language. Sermuga Pandian et al. [56] developed AI-based assistance, name MetaMorph, that enables the automated transfer of low-fidelity sketches to higher-fidelity representations. AI assistance design tools are conducted by [48] and [49], providing a cooperative platform for designers to find inspiring images for their projects quickly. An interactive assistance system has been developed by [28] to aid children in learning numbers. In addition, an intelligent assistance framework proposed by [36] has been designed to provide comprehensive support to pregnant women in rural areas.

5.4 AI Tools

Tools refer to a wide range of resources and methods that are essential for the practice of interaction design [84]. 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 [84].

Some researchers have implemented AI tools such as [5]identifying security threats and [7] for identifying privacy and security threats. A debugging tool is implemented for electronic design assignments in [38]. A web-based tool designed to explain machine learning classification for beginners without needing technical expertise [66]. Developers have adopted the Scratch tool to teach students aged 16-18 AI [17]. Some tools were designed especially for visualizations, such as [39] for teaching computer security. In an interactive environment, the playful authoring tool was implemented to help users create experiences [44]. Atilola et al. [54] implemented a sketch recognition tool called "Mechanix". This tool uses artificial intelligence to detect the shapes and features of a free-body diagram (FBD) sketched into it.

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 [85]. The knowledge base (KB) is defined as a structured database that includes groups of facts [86]. Researchers have used different ways to store knowledge. The knowledge base is stored in text files containing Prolog predicates both Facts and rules [23]. 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. Dialogue manager is a response to provide an appropriate answer to the user by matching the input with KB.

Coding Tutor has three different databases to support students in learning [52]. 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 MySQL database stores elderly answers and questions the system prepared previously for interaction and advice [62]. Questions is designed as seven main questions which focused on three categories physical, mental and sociality. MySQL is adopted due to its popularity as an open-source database management system, which provides efficient data storage and retrieval. It also had the added benefits of scalability, security, and reliability, making it the perfect choice for the caregiver system.

The spoonacular API [1] is used as an ontology to find recipes for each inquiry [45]. The knowledge base of some recipes is collected from AllRecipes[2] website [25]. Also, for the Smart Kitchen, recipes information is collected from the two websites Spoonacular and FoodEssentials [3] [46]. Similarly, in the recommending recipes based on the ingredients system, the dataset is provided by the Allrecipes website [27]. The dataset collected in [56] consists of hand-written sketches.

The AIDA project contains two different conversational agents, each with its own database of questions and answers [65]. The text-based chatbot has a broad spectrum of information about diabetes, while the voice-based dialogue system shows recipes compliant with a diabetic patient's diet. Both databases contain hundreds of questions and answers related to diabetes care and management. The database in May AI [48] is dynamic. The feedback from the designer is used to update the probability distribution of each suggestion agent. A cloud database is used in [36] to allow doctors, gynaecologists, and other social organizations to monitor the cases of pregnant women.

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 [55]. This system of organizing 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. The dataset is collected from the Kisan Call Center (KCC), which is a centralized office used for the purpose of managing and responding large volume of requests by telephone [31, 32].

6.2 Algorithms and Implementation

An Algorithm is defined as, "the thing that gets data processing and other computation done" [87]. 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 7. Our taxonomy is summarized in Table 6.

6.2.1 Conversational AI

This section is further divided based on the domain, with different types of algorithms used to implement chatbots:

- **Health and Agriculture domain** Miura et al. [62] applied to a rule-based virtual caregiver system. The chatbot asks questions elderly, and then the elderly respond, and then the chatbot makes advice based on the answers of the elderly. Momaya et al. [31] used RASA X to implement Farmer Chatbot. The chatbot is developed based on AI and ML techniques.
- **Cooking domain** Chu [45] Used Google Dialogflow to implement the chatbot. Dialogflow is a conversational tool developed by Google with the aim assists software developers in designing chatbots and integrating them into their applications, such as mobile phones or websites [45]. Dialogflow helps developers by allowing them to predefined matching intents by natural language processing and use machine learning to identify patterns [88].

  Similarly, Alloatti et al. [65] provided the Artificial Intelligence Diabetes Assistant (AIDA), which consists of two chatbots, a text-based one for providing diabetic patients with information about diabetes, and a voice-based one for providing patients with recipes compliant with their diet. The architecture of the implementation of AIDA Chatbot is composed of different components; the backend includes Natural Language Understanding (NLU) to interpret user input, Natural Language Generation (NLG) and a Dialogue Manager to ensure appropriate responses to specific queries. Additionally, it uses both a Machine Learning and a Rule-Based engine to generate different questions that extend the language model; providing the neural model with data will further enrich each output with different questions. The aim of this is to provide diabetic patients with a comprehensive AI assistant to aid them in managing their condition.

  The system is designed in [25] as a user enters the ingredients and, based on python implementation and a machine learning algorithm. Upon entering ingredients, the system provides a recipe as its response. Using

[1] *https://spoonacular.com/food-api*
[2] *https://www.allrecipes.com/*
[3] *http://developer.foodessentials.com/*

the IBM Watson platform by Angara et al. [46] to build a conversational kitchen assistant named "foodie fooderson", it allows users to ask by tapping text or voice and retrieves information from a database.

- **Education domain** The chatbot in [52] 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 [34] 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. Shekhar et al. [26] used Rasa to implement the virtual teaching assistant (VTA). RASA is free and open-source for creating virtual assistant (AI) software [26]. A machine algorithm was applied to build Rasa, and the NLU model was used and able to understand the intent of the student's input. The interaction between students and the system is based on messages, and RASA core interprets these messages.

6.2.2 Systems

The AI Doctor system [23] is designed based on facts and rules and implemented with the help of recursive rules; that is, the system checks each fact in its KB against the user's query and, if a match is found, the inference engine presents the appropriate advice or output. If no match is found, the process of checking facts against the query is repeated until the desired answer is found by traversing the decision tree. This system offers the convenience of being able to build the system without requiring programming skills, which is one of its most beneficial features.

Khilji et al. [27] developed a system that combines a rule-based technique and machine learning to classify questions and extract answers based on recommended recipes. The dataset is searched using the keywords retrieved from the questions users ask, and three algorithms are used in the system: a Question Classification (QC) algorithm for training the model; a Recipe Recommendation algorithm that extracts keywords from the question and returns the recipe recommendation; and an Answers Extraction algorithm that takes questions based on types and tags and returns the recipe.

Obeid et al. [55] proposed that integrating an ontology-based recommender system with machine learning could help alumni students choose their major after high school by collecting data when students fill out their profiles, as well as data from a survey that helps students become aware of their interests. Machine learning algorithms, such as K-mode, self-organizing map and hierarchical clustering, are then used to filter and cluster the collected data before it is used to generate the recommendations.

By leveraging the intelligent reading assistant system proposed in [40], 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 utilizing 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 personalized 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 [29]. 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. Sun et al. [30] leveraged AI techniques such as decision tree algorithm and neural networks to design a platform for teaching English.

Researchers in [41] have 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 [57] was enhanced by utilizing 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.

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 [17]. 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 engineering students in learning truss analysis and diagrams [54]. 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 utilized to recognize and compare trusses. The system has a built-in voice recognition module that converts a user's sound into text [63]. A Natural Language Understanding (NLU) module then analyzes the text. A keyword matching module is used to recognize, respond and adapt to different scenarios depending on the words used by the learner. To store and analyze data, the learning system is developed by Amazon Web Services (AWS) lambda functions, which store data in Amazon S3. Additionally, Amazon Comprehend is used to further analyze the data and extract meaningful insights.

6.2.4 AI assistants

Sermuga Pandian et al. [56] used a supervised deep-learning model to create MetaMorp, which enables the model to learn to identify and classify collected dataset, and to generate high-level sketches with better accuracy. ImageSense[49] is implemented as a web application; The user interface is designed utilizing common web-based technologies such as HTML, CSS, and JavaScript, and the back end is implemented by Python framework and database. Google Vision object recognition and semantic labelling API were used to retrieve the semantic label of images. When a designer adds a new image to the board, the Google Vision object recognition and semantic labelling API is used to retrieve the semantic labels of the image and attach the top ten labels to the image. Moreover, the MMCQ algorithm is exploited to analyze each image for its ten major colours.

May AI [48] uses the Cooperative Contextual Bandits (CCB) algorithm, a machine learning framework, to offer or suggest pictures or text that are relevant to the domain. The algorithm uses data from users to make decisions based on expected probabilities for each option. To provide better suggestions for users, the agent can either exploit its current suggestion if it provides the highest probability or explore other options by referring to another agent if that has a higher probability. The algorithm is used in Smart Care [36] to generate personalized dietary guidelines for expecting mothers based on their personal information and situation. It will also provide the ability to track the mother's health condition by comparing it to normal ranges, generating reports, and providing feedback for doctor visits.

Table 6: Summarized Taxonomy used in Table 7, Starting from the Second Column as the First Column is Already Included in Table 7

	Taxonomy	Description
2	Knowledge store	Database (DB), Knowledge base (KB), Dataset (DS), Text files (TF), Folder (FO)
3	Knowledge update	Static (S), Dynamic (D)
4	Entering input	Patients (P), Users (U), Learners (L), Novice Programmers (NP), Elderly People (EP), Security Developers (SD), Instructors (I), Children (CH), Designer (D), Pregnant Women (PW)
5	Algorithms or methods	Decision tree algorithm (DTA), Machine Larning (ML),Generate guidelines and Monitor health (GG and MH), Natural Language Algorithm (NLA), Recommendation Algorithm (RA), Deep Neural Networks (DNNs), Modified Median Cut Quantization(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), Recognizing Truss (RTruss), Recursive (Rec), Lambda (LAM)
6	Types of interaction	Written (W), Spoken (S), Action (A), Visualization (V)

Table 7: Implementation of Different Approaches in AI Techniques

Names of system/tool	**knowledge store**	**knowledge update**	**Entering input**	**Algorithm/method**	**Interaction types**
(1)	(2)	(3)	(4)	(5)	(6)
MetaMorph [56]	DS	—	D	DNNs	V
ImageSense [49]	DB	—	D	NLA, MMCQ	W,V
May AI [48]	DB	D	U	CCB	V
iList [57]	—	D	U	ML	W,V
Smart Care [36]	DB	D	PW	GG, MH	W,V
Assistance System [28]	DB	D	CH	IR	V
AI Doctor [23]	TF	S	P	DTA	W
Coding Tutor [52]	DB	D	NP	—	W
Caregiver system [62]	DB	S	EP	—	W
Foodie Fooderson [46]	DB	D	U	—	W,S
AIDA [65]	KB	S	P	—	W,S
HealthAssistantBot [47]	DB	S	P	—	W
Alexa for E-learning[63]	—	—	U	LAM, NLA	S
Microsoft Threat Modelling Tool [5], OWASP [7]	—	—	SD	STRIDE	W,V
C'Meal[25]	FO	S	U	ML	W
Recipe Bot [45]		S	U	ML	W,S
An Intelligent Reading Assistant System [40]	DB	—	L	RA	W
Intelligent English Teaching Platform [30]	NN	—	I,L	DTA, NN	W
The Toastboard [38]	—	—	U	—	A
Teachable Machine [66]	—	—	U	ML	V

Table 7: Continued

Names of system/tool	knowledge store	knowledge update	Entering input	Algorithm/method	Interaction types
(1)	(2)	(3)	(4)	(5)	(6)
Zhorai [67]	—	—	CH	—	W,V
A Scratch-based Artificial Intelligence tutorial [17]	—	—	L	CL, NN	V
Smartpathk [41]	—	D	I,L	SML, J48	W,V
Chatbot [34]	—	—	L	—	W
An Intelligent Assistant [24]	—	—	L	—	W
Visualization Tools [39]	—	—	L	—	W,V
VTK [43]	—	—	L	—	W,V
Playful Authoring Tools [44]	—	—	U	—	W,V,S
Mental Vision [42]	—	—	L	—	W,V
Intelligent Tutoring [80]	—	—	L	—	W,V
Mechanix [54]	—	—	L	GBA, RTruss	W,V
Krushi [31]	KCC	—	U	Rasa X	W,V
Trigger-action-circuits [29]	DB	—	U	BFS, Rec, DRA	A
CVTA [26]	—	—	L	ML	W
HeyTeddy [35]	—	D	NP	—	W,S,A
ThinkInk[37]	—	—	U	RATA	W,V

6.3 Evaluation

In this section, we will thoroughly examine the evaluation methods used by each paper to evaluate the AI techniques, with a specific focus on the question "How do they evaluate?", as indicated in Table 9, most papers utilised experimental evaluation. 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. Some papers did not undergo evaluation, citing reasons such as being in a developmental stage such as [23] or serving as conceptual or introductory pieces. Certain papers outlined the design and functionality of the AI technique without delving into empirical evaluation [45]. In Table 8, we categorized the AI techniques to analyse the implementation of experimental evaluation. Our analysis indicates that a majority of AI techniques incorporated both quantitative and qualitative methods.

Table 8: Methods Classified by Utilized Methodology Classification of AI Techniques Based on Employed Methodology, including Qualitative, Quantitative, and Mixed Approaches

	Chatbot	System	Tools	AI assistance
Qualitative				[24],[64]
Quantitative	[31]	[57]	[67]	
Mixed	[62], [47], [34], [35]	[52],[29],[32]	[38],[39],[17],[54],[37]	[56],[48],[69]

6.3.1 Challenges and Limitations in AI Techniques

- **Challenges in Engagement and User Interaction** Challenges in engagement and user interaction present a common thread in several AI techniques, revealing underlying difficulties faced by users across various systems and applications. For example, some elderly participants in [62] discontinued services due to challenges related to operation and limited smartphone use, coupled with health issues that impacted their ability to engage with chatbot questions. This resonates with the chatbot [34], where users expressed limitations and dissatisfaction with a chatbot's responses, further demonstrating how complex or unintuitive interaction design can hinder user engagement. In [48], some users found it challenging to understand AI's suggestions in the

Table 9: Evaluation of AI Approaches: Chatbots denoted by white background, AI systems by blue, AI Tools by green, and AI assistants by grey. Dash (-) signifies studies without findings

Ref.	Eval. types		Data collection methods	Participants	Metrics/Measures
	Exp	Theo			
[62]	✓		Interviews and questionnaires	8 elderly and 19 younger	Responsiveness and continuity, mental state data and quality in use, effectiveness, efficiency, usefulness, comfort, and flexibility.
[65]	-	-	-	-	-
[47]	✓		Questionnaires	102 participants	Accuracy testing(Algorithm) and user evaluation satisfaction and the ease of interaction.
[46]	-	-	-	-	-
[34]	✓		-	14,000 people	User-chatbot interaction and usability.
[45]	-	-	-	-	-
[35]		✓	Sessions with participants and Interviews	10 users	Time, debugging issues and users' confidence
[26]	-	-	-	-	-
[31]	✓		Web scraping	-	Accuracy
[25]	-	-	-	-	-
[57]	✓		Tests for Learning Assessment, surveys, student actions record	between 214 and 219 students	Learning, satisfaction, and problem-solving behavior
[53]	-	-	-	-	-
[52]	✓		Demonstrations, self assessments, programming tasks, questionnaires, written feedback	40 students	Usability, effectiveness, efficiency, and acceptance among its intended users.
[53]	-	-	-	-	-
[27]	-	-	-	3 participants	Efficiency of the proposed method and the performance of the system.
[40]		✓	-	-	Reading ability
[29]	✓		Complete a task: building a circuit using an Arduino with specific time	12 participants	Usability, time and efficient of the system
[41]	-		-	-	The accuracy of the model generated, the comprehensive ability of the extracted knowledge, the learning time
[63]	-	-	-	15 learners	The improvement in the learners' interviewing skills after using the application.
[80]	-	-	-	-	-
[38]	-	-	Interviews and observations	7 participants	
[67]	-	-	Pilot Study and user study	14 children	The effective engagement of Zhorai and the children's understanding
[39]	✓		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.
[17]	✓		Questiners and open questions	37 students	Understanding of AI
[66]	-	-	-	-	-
[54]	✓		Assessment and focus group	70-100 students (regular), 30-40 (honors).	The effectiveness of Mechanix
[42]	-	-	-	-	-
[37]	✓		Likert-scale questionnaire	45 students	Interactivity, design, playfulness, ease of use, usefulness, and intention to use.
[43]	-	-	-	-	-
[48]	✓		Standardized measurements and semi-structured interviews	16 professional designers	The effectiveness and impact of an AI tool on the creative design.
[28]	✓		Assessing children's progress in learning numbers	Group of six children	The progress and performance of children in learning numbers.

Table 9: Continued

Ref.	Eval. types		Data collection methods	Participants	Metrics/Measures
	Exp	Theo			
[24]	✓		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.
[36]	-	-	-	-	-
[69]	✓		Multiple-choice questions and pen questions	494 children	knowledge improvement, perception of Learning ML, and changes in the perception of AI.
[64]	✓		Assessing task outcomes, 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.

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 [64], challenges adapting to and effectively utilizing tutoring systems were reported, indicating that these barriers are not confined to one area of technology but extend across various domains. These examples collectively illustrate a recurring challenge in designing interfaces and systems that are both user-friendly and capable of meeting diverse user needs and expectations, reflecting a fundamental concern in the field of human-computer interaction.

- **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 [47], challenges emerge with the underperformance of the MLP model and difficulties in making definitive conclusions due to statistical and algorithmic constraints. These include the interchangeability of algorithms, a decrease in performance with more ensemble models, and difficulties with complex and vague conditions like bipolarism and PTSD. The study in [35] further compounds these limitations by identifying issues in handling complex prototypes, lack of multi-user support, and challenges in integrating new features. Meanwhile, [31] points to non-compliance with local languages and ineffective sound interaction, signaling a need for retraining and highlighting underlying inadequacies in current systems. In [52], the diverse programming experience among users, the need for software improvements, and server capacity limitations reflect struggles with adaptability and responsiveness. [29] reveals further challenges in optimization, limited support for higher-level software constructs, and difficulties in supporting circuits without a microcontroller, with a focus only on novice users. Finally, [38] illustrates constraints in handling complex circuit debugging scenarios, including a limited ability to capture rapidly changing values and analyze embedded software behavior, indicating a need for new solutions. Together, these examples paint a picture of a technological landscape where models, tools, and systems often fall short in handling complexity, adaptability, and specificity. This impacts not only efficiency and accuracy but also challenges our ability to create universally relevant and robust technologies.
- **Challenges in Education and Learning Tools** The recurring theme across the references pertains to the challenges and limitations faced in educational technology and research methodologies. Reference6 emphasizes the dissatisfaction with repetitive feedback in a specific version of iList, linking with the overall theme of user experience and feedback. References 11, 12, 13, 15, and 19 share a common concern regarding small sample sizes, limiting the strength and generalizability of conclusions, and pointing to the need for more extensive research and controlled interaction. This reflects a broader challenge in educational research, as mentioned in reference20, where adaptation to new tools like the SPA tutor might present limitations compared to human interaction. Reference18 touches on user experience again, this time focusing on the unique challenges faced by young children in terms of technical complexity and cognitive load. The common thread that runs through these references is the complex interplay between technology, user experience, and research methodology, illustrating the multifaceted challenges in creating and evaluating effective educational tools and experiences.
- **Challenges in AI Education and Learning Techniques** The study [57] emphasizes dissatisfaction with repetitive feedback in a specific version of iList, linking it with the overall theme of user experience and feedback. References [67], [39], [17], [37], and [69] 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 [64], where adaptation to new tools like the SPA tutor may present limitations compared to human interaction. Researchers in [28] 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 common thread that runs through these references

is the complex interplay between technology, user experience, and research methodology, illustrating the multifaceted challenges in creating and evaluating effective educational tools and experiences.

- **Limited Sample Size Affecting Conclusions:** challenges in some sources, which limits the strength of the claim regarding the effectiveness of Zhorai [67]. The study involving Zhorai also highlights the need for future research to compare the conversational agent interface with other interfaces, such as a text-based interface, and mentions that controlled interaction, including limited knowledge of ecosystems, was necessary but could affect the experience. The results of student questionnaires and educational visualization tools are promising in[39], but limited sample sizes and one-time workshops make it challenging to draw definitive conclusions, emphasizing the need for more extensive studies, sophisticated materials, and alternative programming languages [17]. Additionally, limited availability, potential bias in selection, and curriculum changes present challenges in evaluation [54]. Individual biases and interpretations, the time-consuming nature of data collection, and the possibility of socially desirable answers instead of true feelings reveal inherent complexities in methodology [37]. The reduction in sample size, potential influence of prior knowledge, and gender disparities in STEM also may affect the generalizability of findings [37]. Finally, the challenges for users, such as adapting to new tools and methodologies, encapsulate the multifaceted nature of these challenges [64].
- **User Dissatisfaction and Experience Challenges:** Some elderly users faced difficulties in operating and engaging with a chatbot [62], while other users were dissatisfied with app functions and chatbot interactions [34], illustrating the critical need for user-centric design. Feedback issues caused dissatisfaction among students, with one version being viewed as more repetitive and misleading than others [57], emphasizing the importance of clear, varied feedback. Users also struggled with understanding an AI system's suggestions, leading to concerns about its complexity and relevance [48]. The challenges with adapting to an automated tutor, compared to a human tutor [64], demonstrate potential limitations in technology's ability to mimic human-like understanding and responsiveness. Together, these ideas underscore the importance of aligning technological systems with user expectations and needs, and investing in iterative design and user support to enhance satisfaction and experience.
- **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. Challenges range from elderly participants struggling with technology operation [62], to intricate challenges in medical modeling [47], design limitations in handling prototypes [35], language processing issues [31], debugging difficulties [29], and complexities in circuit debugging [38]. Learning curve and technical limitations in the evaluation process [56], user understanding of AI-driven processes [48], and technical complexity for children's interaction [28] further emphasize the multifaceted nature of these challenges. Whether 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 multifaceted solutions that consider users' needs, technical capabilities, and the specific context in which the system operates.

6.4 Types of Interactions Users

In order for users to interact with AI techniques, the interaction can be divided into four types: text, voice, visualization, and action. This allows users to access the machine and receive accurate responses to their queries.

6.4.1 Interaction Based on Writing

Interaction based on text is a form of communication between end user and an AI system, where the user interacts with the AI system through written text. This form of communication makes it possible for users to interact with AI systems in real-time, allowing for more natural and effective conversations. Interaction 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 Visulization

The use of visualization in the interaction between users and AI tools has numerous benefits. Visualization helps to quickly convey complex information to users in an easier to understand. Additionally, visualization can also help to better identify potential issues, allowing users to make informed decisions more quickly. For example, as mentioned earlier in Section 3.1, when discussing Microsoft's threat model and Threat Dragon [5, 7]. Also, visualization of the output helps to speed up the process 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

Table 10: Interaction Types (Text, Sound, Action, and Visualizations) in Academic Research Papers

Interaction Types	Number of papers
Writen	47%
Spoken	10%
Action	11%
Visualization	32%

groups [67], which revealed an increase in engagement during learning due to the conversational aspect. Utilizing a combination of conversation and visuals was successful in educating kids on the fundamentals of machine learning. The use of visualization 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; rather, it has been further augmented to include voice interactions to meet the needs of users. Terzopoulos and Satratzemi [89] mentioned some intriguing features of voice assistants. The most used activities include: responding to questions posed by users, streaming music from online services, creating alarms and timers, playing games, making calls or sending messages, completing purchases, and supplying weather. Researchers [35] mentioned that using voice-over typing in conversational interfaces has more advantages than typing—such as making programming easier for novices programming. Some studies such as [35, 63, 64] have used Alexa as a visual assistant. To access Alexa, an Amazon Echo Dot device is used to handle the voice and then convert the voice input to text [35], and Dizon [70] used intelligent personal assistants (Alexa) to educate English.

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, beginners programmer can learn by building and assembling circuits, as evidenced by the works of Anderson et al. [29], Kim et al. [35], Drew et al. [38].

To summarise, The majority of interactions between users and AI techniques are based on written at 47%. Visualization accounts for 32%, while action and spoken interactions make up 11%, and 10%, respectively, as shown in Table 10. However, we cannot generalise this result as a fact because this is based on and restricted to the literature review. Furthermore, some papers combine different types of interaction, while some papers only include one type for their purpose.

7 RESEARCH CHALLENGES AND DIRECTIONS

AI techniques are gaining increasing prominence across various domains, offering explicit or implicit learning capabilities through the teaching process. However, teaching privacy by design remains a formidable challenge. In this literature review, we identify several research challenges and propose directions for the development of a technique that effectively teaches privacy and laws to novice software engineers.

7.1 Lack of Techniques for Educating about Privacy and Laws

Previous research has introduced numerous AI techniques, such as machine learning algorithms and natural language processing, to enhance the knowledge of beginners and enable users to develop expertise in specific domains like healthcare, education, and cooking. While some tools, such as the Microsoft threat modeling tool and Threat Dragon, have proven valuable for identifying threats during system development, they are not designed to assist novice developers or serve as educational aids for designing Internet of things applications. Furthermore, IoT applications often involve the collection and analysis of personal data, which is frequently sensitive. Consequently, data privacy laws exist to safeguard this information. The main issue identified here is that non-functional requirements, like privacy, are frequently overlooked due to the inherent complexity of IoT applications. It has been proven that PbD is beneficial for introducing privacy at an early stage in the design development life cycle, thus minimizing privacy issues and threats [90][4]. However, novice software developers often find it challenging to incorporate privacy into their designs. Therefore, they need to be educated about privacy, either implicitly or explicitly, using techniques that can facilitate this. [4],[90]. Novice software developers find it difficult to apply privacy into their designs. Therefore, they need be educated about privacy either implicitly or explicitly which AI techniques could help to do so.

While some studies, such as the one conducted by Vaidya et al. [91], propose innovative approaches to teaching privacy alongside traditional methods, our focus is on leveraging the PbD approach. This approach encompasses various elements, including privacy principles, guidelines, strategies, goals, and patterns. By implementing practical use cases using our techniques, novices can gain a better understanding of PbD and privacy laws.

7.2 The Need for an Intelligent Technique

In recent years, there has been a discernible and noteworthy trend cutting across various fields: the increasingly pervasive integration of AI techniques into the development of intelligent software systems, as highlighted by Rodrigues [92]. Notably, key domains such as Audio Processing, Natural Language Processing, Information Retrieval, and Computer Vision have borne witness to substantial advancements that stand poised to fundamentally transform our interactions with machines [92].

However, developing an intelligent technique capable of effectively capturing user-manipulated data during the design process and furnishing pertinent insights concerning potential application threats remains a formidable challenge. Our overarching vision is to engineer an intelligent technique that can guide for novices in the realm of PbD, while also possessing a host of intelligent features, as delineated in this comprehensive literature review. These features encompass real-time feedback, interactive question-answering capabilities, and adaptability to a multitude of scenarios. The development of such an AI-mediated technique may hold the potential to greatly expedite the learning curve for fledgling software engineers.

7.3 The Need for Effective Teaching Strategies

In order to effectively educate novice software engineers about privacy by design, it is crucial to develop a teaching technique that employs effective strategies. These strategies should be customized to the engineers' level of understanding, prior knowledge, and interests. Furthermore, they should be designed to ensure user engagement and motivation for learning. Interactive approaches, such as highlighting privacy threats and proposing corresponding privacy-preserving measures in specific contexts, can significantly enhance the learning experience. Additionally, the technique should provide immediate feedback to users by listing all identified threats along with proposed suggestions.

To further enhance the learning process, the application of Bloom's taxonomy [93] can be invaluable. It assists in creating a learning framework that may help novices remember, understand, and apply privacy principles and laws in their design applications. Some studies have already successfully applied Bloom's taxonomy in computer science education [94] and in software engineering education [95], [96].

7.4 The Need for Focusing on Education for Novice Software Engineers

Novice software engineers are individuals new to the field, either as students or recent entrants into the professional sphere. It is crucial to enhance their knowledge about privacy because, even for experienced software engineers, the concept remains vague [4]. By concentrating on novice software engineers, the primary objective is to facilitate their early acquisition of knowledge in preventing potential privacy risks, as opposed to the necessity of addressing the repercussions of privacy violations post-application launch. Violating data privacy laws such as the GDPR [97] could result in severe fines or penalties, underlining the importance of early and effective privacy education.

Perry and Roda [98] provided a curriculum specifically designed for teaching non-technical audiences about privacy by design. This includes introducing them to concepts such as data protection, understanding how to use privacy by design, and emphasizing the importance of building privacy into products. Also, they encouraged discussion among students in order to foster critical thinking skills related to security and privacy topics. This involved having students discuss their own experiences with data protection or discussing the implications of different types of technology on our lives. Conrad [99] provided the NIST privacy framework to educate APP developers on the most effective way to put Privacy by Design into practice. However, there are no techniques aimed and focused on helping beginner software engineers to understand privacy during the design of IoT applications.

Our focus squarely rests on educating novice software engineers, those fresh to the software engineering domain, including recent graduates and newcomers. These engineers typically lack the experience and depth of understanding possessed by their more seasoned counterparts. Therefore, they require comprehensive guidance and support to navigate the complexities of integrating privacy considerations early in the software development lifecycle

8 CONCLUSIONS

This survey has discussed some design tools and characteristics in the previously existing literature. In addition, This paper proposed a taxonomy for AI techniques in learning and how users interact with systems to increase their awareness. The learning is divided into two types explicit and implicit. It is evident that users gain benefits and enhance their knowledge implicitly or explicitly when using AI techniques. Furthermore, the implementation of these AI assistant techniques has been discussed. Although most AI assistant techniques can help users learn implicitly or explicitly by performing tasks with these tools, some techniques lack detailed information about algorithms or how to update their knowledge. Additionally, this survey has elucidated how prior studies evaluate AI techniques and has unveiled challenges encountered during the evaluation process. The study has also brought attention to potential research hurdles, proposing solutions that warrant consideration in the future.

References

[1] Aloysius Henry Sequeira. Introduction to concepts of teaching and learning. *Available at SSRN 2150166*, 2012.

[2] Elina Riivari, Marke Kivijärvi, and Anna-Maija Lämsä. Learning teamwork through a computer game: for the sake of performance or collaborative learning? *Educational Technology Research and Development*, 69:1753–1771, 2021.

[3] Binyang Song, Nicolas F Soria Zurita, Hannah Nolte, Harshika Singh, Jonathan Cagan, Christopher McComb, et al. When faced with increasing complexity: The effectiveness of ai assistance for drone design. *JOURNAL OF MECHANICAL DESIGN*, 143:1–38, 2021.

[4] Nada Alhirabi, Omer Rana, and Charith Perera. Designing security and privacy requirements in internet of things: A survey. *arXiv preprint arXiv:1910.09911*, 2(6):1—–37, 10 2019. URL `http://arxiv.org/abs/1910.09911`.

[5] Microsoft. Microsoft threat modelling, October 2020. URL `https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling`.

[6] Linddun. Linddun privacy threat modeling, October 2014. URL `https://www.linddun.org/`.

[7] OWASP. Owasp threat dragon, oct 2020. URL `https://owasp.org/www-project-threat-dragon/`.

[8] Amy L Baylor and Yanghee Kim. Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education*, 15(2):95–115, 2005.

[9] Claes Wohlin. 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*, pages 1–10, 2014.

[10] Jeroen van Rest, Daniel Boonstra, Maarten Everts, Martin van Rijn, and Ron van Paassen. Designing privacy-by-design. In *Privacy Technologies and Policy: First Annual Privacy Forum, APF 2012, Limassol, Cyprus, October 10-11, 2012, Revised Selected Papers 1*, pages 55–72. Springer, 2014.

[11] Richmond Y Wong and Deirdre K Mulligan. Bringing design to the privacy table: Broadening "design" in "privacy by design" through the lens of hci. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–17, 2019.

[12] Stephan Seifermann, Robert Heinrich, Dominik Werle, and Ralf Reussner. Detecting violations of access control and information flow policies in data flow diagrams. *Journal of Systems and Software*, 184:111138, 2022. ISSN 0164-1212. doi: https://doi.org/10.1016/j.jss.2021.111138. URL `https://www.sciencedirect.com/science/article/pii/S0164121221002351`.

[13] Daniele Granata, Massimiliano Rak, and Giovanni Salzillo. Automated threat modeling approaches: Comparison of open source tools. In *International Conference on the Quality of Information and Communications Technology*, pages 250–265. Springer, 2022.

[14] Imano Williams and Xiaohong Yuan. Evaluating the effectiveness of microsoft threat modeling tool. pages 1–6. Association for Computing Machinery, Inc, 10 2015. ISBN 9781450340496. doi: 10.1145/2885990.2885999.

[15] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4):1–15, 2010.

[16] Antonio Sarasa-Cabezuelo. Use of scratch for the teaching of second languages. *International Journal of Emerging Technologies in Learning (iJET)*, 14(21):80–95, 2019.

[17] Julian Estevez, Gorka Garate, and Manuel Graña. Gentle introduction to artificial intelligence for high-school students using scratch. *IEEE access*, 7:179027–179036, 2019.

[18] Mohammed Z Shaath, Mones Al-Hanjouri, Samy S Abu Naser, and Rami ALdahdooh. Photoshop (cs6) intelligent tutoring system. *International Journal of Academic Research and Development*, 2(1), 2017.

[19] Morgan Benton. The knowledge illusion: Why we never think alone. *Software Quality Professional*, 19(4):45, 2017.

[20] Youji Kohda. Can humans learn from ai? a fundamental question in knowledge science in the ai era. In *Advances in the Human Side of Service Engineering: Proceedings of the AHFE 2020 Virtual Conference on The Human Side of Service Engineering, July 16-20, 2020, USA*, pages 244–250. Springer, 2020.

[21] Esther Ziegler, Peter A Edelsbrunner, and Elsbeth Stern. The relative merits of explicit and implicit learning of contrasted algebra principles. *Educational Psychology Review*, 30(2):531–558, 2018.

[22] Rod Ellis. 1. implicit and explicit learning, knowledge and instruction. In *Implicit and explicit knowledge in second language learning, testing and teaching*, pages 3–26. Multilingual Matters, 2009.

[23] Sumit Das, S Biswas, Aditi Paul, and Aritra Dey. Ai doctor: An intelligent approach for medical diagnosis. In *Industry Interactive Innovations in Science, Engineering and Technology: Proceedings of the International Conference, I3SET 2016*, pages 173–183. Springer, 2018.

[24] Mario Konecki, Nikola Kadoić, et al. Intelligent assistant for helping students to learn programming. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 924–928. IEEE, 2015.

[25] Alessandra Amato and Giovanni Cozzolino. C'meal! the chatbot for food information. In *International Conference on Intelligent Networking and Collaborative Systems*, pages 238–244. Springer, Springer, 2020.

[26] Gaurav Shekhar, Rhea D'Souza, and Kevin Fernandes. Ai-driven contextual virtual teaching assistant using rasa. page 346. Association for Computing Machinery, Inc, 10 2020. ISBN 9781450370455. doi: 10.1145/3368308.3415442.

[27] Abdullah Faiz Ur Rahman Khilji, Riyanka Manna, Sahinur Rahman Laskar, Partha Pakray, Dipankar Das, Sivaji Bandyopadhyay, and Alexander Gelbukh. Cookingqa:answering questions and recommending recipes based on ingredients. *Arabian Journal for Science and Engineering*, 46(4):3701–3712, 2021.

[28] William Villegas-Ch, Angel Jaramillo-Alcázar, and Aracely Mera-Navarrete. Assistance system for the teaching of natural numbers to preschool children with the use of artificial intelligence algorithms. *Future Internet*, 14(9): 266, 2022.

[29] Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 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*, pages 331–342, 2017.

[30] Zhuomin Sun, M Anbarasan, and DJCI Praveen Kumar. Design of online intelligent english teaching platform based on artificial intelligence techniques. *Computational Intelligence*, 37(3):1166–1180, 2021.

[31] Mihir Momaya, Anjnya Khanna, Jessica Sadavarte, and Manoj Sankhe. Krushi–the farmer chatbot. In *2021 International Conference on Communication information and Computing Technology (ICCICT)*, pages 1–6. IEEE, 2021.

[32] Mohit Jain, Pratyush Kumar, Ishita Bhansali, Q Vera Liao, Khai Truong, and Shwetak Patel. Farmchat: a conversational agent to answer farmer queries. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(4):1–22, 2018.

[33] Naman Jain, Pranjali Jain, Pratik Kayal, Jayakrishna Sahit, Soham Pachpande, Jayesh Choudhari, et al. Agribot: agriculture-specific question answer system. 2019.

[34] Xuan Lam Pham, Thao Pham, Quynh Mai Nguyen, Thanh Huong Nguyen, and Thi Thu Huong Cao. Chatbot as an intelligent personal assistant for mobile language learning. In *Proceedings of the 2018 2nd International Conference on Education and E-Learning*, pages 16–21, 2018.

[35] Sang Gook Kim, Sang Min Yoon, M. Yang, Jungwoo Choi, H. Akay, and Edward Burnell. Heyteddy: Conversational test-driven development for physical computing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3, 2019. ISSN 24749567. doi: 10.1145/3369838.

[36] Sanjida Nasreen Tumpa, Anika Binte Islam, and Md Tasnim Manzur Ankon. Smart care: An intelligent assistant for pregnant mothers. In *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, pages 754–759. IEEE, 2017.

[37] Md Athar Imtiaz, Andrew Luxton-Reilly, and Beryl Plimmer. Thinkink-an intelligent sketch tool for learning data structures. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2018.

[38] Daniel Drew, Julie L Newcomb, William McGrath, Filip Maksimovic, David Mellis, and Björn Hartmann. The toastboard: Ubiquitous instrumentation and automated checking of breadboarded circuits. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 677–686, 2016.

[39] Xiaohong Yuan, Percy Vega, Yaseen Qadah, Ricky Archer, Huiming Yu, and Jinsheng Xu. Visualization tools for teaching computer security. *ACM Transactions on Computing Education (TOCE)*, 9(4):1–28, 2010.

[40] Jianxuan Zhao, Huazhen Wang, Xiaofeng Wang, and Jin Gou. The research and implementation of intelligent reading assistant system framework based on knowledge graph. In *2021 The 3rd World Symposium on Software Engineering*, pages 45–51, 2021.

[41] Nayze Lucena Sangreman Aldeman, Keylla Maria de Sá Urtiga Aita, Vinícius Ponte Machado, Luiz Claudio Demes da Mata Sousa, Antonio Gilberto Borges Coelho, Adalberto Socorro da Silva, Ana Paula da Silva Mendes, Francisco Jair de Oliveira Neres, and Semíramis Jamil Hadad do Monte. Smartpathk: a platform for teaching glomerulopathies using machine learning. *BMC medical education*, 21(1):1–8, 2021.

[42] Achille Peternier, Daniel Thalmann, and Frédéric Vexo. Mental vision: a computer graphics teaching platform. In *International Conference on Technologies for E-Learning and Digital Entertainment*, pages 223–232. Springer, 2006.

[43] Paulo Dias, Joaquim Madeira, and Beatriz Sousa Santos. Using vtk as a tool for teaching and applying computer graphics. In *Eurographics (Education Papers)*, pages 61–67, 2006.

[44] Casper Harteveld, Nolan Manning, Farah Abu-Arja, Rick Menasce, Dean Thurston, Gillian Smith, and Steven C Sutherland. Design of playful authoring tools for social and behavioral science. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces Companion*, pages 157–160, 2017.

[45] Jiawen Chu. Recipe bot: The application of conversational ai in home cooking assistant. pages 696–700. Institute of Electrical and Electronics Engineers Inc., 2021. ISBN 9781665427098. doi: 10.1109/ICBASE53849.2021.00136.

[46] Prashanti Angara, Miguel Jiménez, Kirti Agarwal, Harshit Jain, Roshni Jain, Ulrike Stege, Sudhakar Ganti, Hausi A Müller, and Joanna W Ng. Foodie fooderson a conversational agent for the smart kitchen. In *CASCON*, pages 247–253. ACM, 2017.

[47] Marco Polignano, Fedelucio Narducci, Andrea Iovine, Cataldo Musto, Marco De Gemmis, and Giovanni Semeraro. Healthassistantbot: A personal health assistant for the italian language. *IEEE Access*, 8:107479–107497, 2020. ISSN 21693536. doi: 10.1109/ACCESS.2020.3000815.

[48] Janin Koch, Andrés Lucero, Lena Hegemann, and Antti Oulasvirta. May ai? design ideation with cooperative contextual bandits. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[49] Janin Koch, Nicolas Taffin, Michel Beaudouin-Lafon, Markku Laine, Andrés Lucero, and Wendy E Mackay. Imagesense: an intelligent collaborative ideation tool to support diverse human-computer partnerships. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–27, 2020.

[50] David Boud and Elizabeth Molloy. Feedback in higher and professional education. *Understanding It and Doing It Well*, page 2013, 2013.

[51] Sharon Oviatt, Alex Arthur, and Julia Cohen. Quiet interfaces that help students think. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 191–200, 2006.

[52] Sebastian Hobert. Say hello to 'coding tutor'! design and evaluation of a chatbot-based learning system supporting students to learn to program. 2019.

[53] Lisa N Michaud, Kathleen F McCoy, and Christopher A Pennington. An intelligent tutoring system for deaf learners of written english. In *Proceedings of the fourth international ACM conference on Assistive technologies*, pages 92–100, 2000.

[54] Olufunmilola Atilola, Stephanie Valentine, Hong-Hoe Kim, David Turner, Erin McTigue, Tracy Hammond, and Julie Linsey. Mechanix: A natural sketch interface tool for teaching truss analysis and free-body diagrams. *AI EDAM*, 28(2):169–192, 2014.

[55] Charbel Obeid, Inaya Lahoud, Hicham El Khoury, and Pierre-Antoine Champin. Ontology-based recommender system in higher education. In *Companion Proceedings of the The Web Conference 2018*, pages 1031–1034, 2018.

[56] Vinoth Pandian Sermuga Pandian, Sarah Suleri, Christian Beecks, and Matthias Jarke. Metamorph: Ai assistance to transform lo-fi sketches to higher fidelities. In *Proceedings of the 32nd Australian Conference on Human-Computer Interaction*, pages 403–412, 2020.

[57] Davide Fossati, Barbara Di Eugenio, STELLAN Ohlsson, Christopher Brown, and Lin Chen. Data driven automatic feedback generation in the ilist intelligent tutoring system. *Technology, Instruction, Cognition and Learning*, 10(1):5–26, 2015.

[58] Sarah Buchanan and Joseph J Laviola Jr. Cstutor: A sketch-based tool for visualizing data structures. *ACM Transactions on Computing Education (TOCE)*, 14(1):1–28, 2014.

[59] Julia Schuchard and Cynthia K Thompson. Implicit and explicit learning in individuals with agrammatic aphasia. *Journal of psycholinguistic research*, 43(3):209–224, 2014.

[60] Arthur S Reber. Implicit learning of artificial grammars. *Journal of verbal learning and verbal behavior*, 6(6): 855–863, 1967.

[61] James A Crowder, John Carbone, and Shelli Friess. Implicit learning in artificial intelligence. In *Artificial Psychology*, pages 139–147. Springer, 2020.

[62] Chisaki Miura, Sinan Chen, Sachio Saiki, Masahide Nakamura, and Kiyoshi Yasuda. Assisting personalized healthcare of elderly people: Developing a rule-based virtual caregiver system using mobile chatbot. *Sensors*, 22 (10):3829, 2022.

[63] Jinjin Zhao, Shreyansh Bhatt, Candace Thille, Dawn Zimmaro, Neelesh Gattani, and Josh Walker. Introducing alexa for e-learning. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pages 427–428. ACM, 2020.

[64] Rainer Winkler, Matthias Söllner, Maya Lisa Neuweiler, Flavia Conti Rossini, and Jan Marco Leimeister. Alexa, can you help us solve this problem? how conversations with smart personal assistant tutors increase task group outcomes. In *Extended abstracts of the 2019 CHI conference on human factors in computing systems*, pages 1–6, 2019.

[65] Francesca Alloatti, Alessio Bosca, Luigi Di Caro, and Fabrizio Pieraccini. Diabetes and conversational agents: the aida project case study. *Discover Artificial Intelligence*, 1(1):1–21, 2021.

[66] Michelle Carney, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, and Alexander Chen. Teachable machine: Approachable web-based tool for exploring machine learning classification. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems*, pages 1–8, 2020.

[67] Phoebe Lin, Jessica Van Brummelen, Galit Lukin, Randi Williams, and Cynthia Breazeal. Zhorai: Designing a conversational agent for children to explore machine learning concepts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13381–13388, 2020.

[68] Juan David Rodríguez García, Jesús Moreno-León, Marcos Román-González, and Gregorio Robles. Learningml: a tool to foster computational thinking skills through practical artificial intelligence projects. *Revista de Educación a Distancia (RED)*, 20(63), 2020.

[69] Juan David Rodríguez-García, Jesús Moreno-León, Marcos Román-González, and Gregorio Robles. 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*, pages 177–183, 2021.

[70] Gilbert Dizon. Using intelligent personal assistants for second language learning: A case study of alexa. *Tesol Journal*, 8(4):811–830, 2017.

[71] Jianfeng Gao, Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Heung-Yeung Shum. Robust conversational ai with grounded text generation. 9 2020. URL `http://arxiv.org/abs/2009.03457`.

[72] RJ Ong, RAA Raof, S Sudin, and KY Choong. A review of chatbot development for dynamic web-based knowledge management system (kms) in small scale agriculture. In *Journal of Physics: Conference Series*, volume 1755, page 012051. IOP Publishing, 2021.

[73] Lee Boonstra. Introduction to conversational ai. In *The Definitive Guide to Conversational AI with Dialogflow and Google Cloud*, pages 1–27. Springer, 2021.

[74] Natalie Wellnhammer, Mateusz Dolata, Susanne Steigler, and Gerhard Schwabe. Studying with the help of digital tutors: design aspects of conversational agents that influence the learning process. 2020.

[75] Hankyung Kim and Youn-kyung Lim. Teaching-learning interaction: a new concept for interaction design to support reflective user agency in intelligent systems. In *Designing Interactive Systems Conference 2021*, pages 1544–1553, 2021.

[76] Edwin D Reilly. Interactive system. In *Encyclopedia of Computer Science*, pages 907–908. 2003.

[77] Nigel Major, Shaaron Ainsworth, and David Wood. Redeem: Exploiting symbiosis between psychology and authoring environments. *International Journal of Artificial Intelligence in Education*, 8(3/4):317–340, 1997.

[78] Antonija Mitrovic. Modeling domains and students with constraint-based modeling. In *Advances in intelligent tutoring systems*, pages 63–80. Springer, 2010.

[79] Bradley Herbert, Mark Billinghurst, Amali Weerasinghe, Barret Ens, and Grant Wigley. A generalized, rapid authoring tool for intelligent tutoring systems. In *Proceedings of the 30th Australian Conference on Computer-Human Interaction*, pages 368–373, 2018.

[80] Kenneth R Koedinger, John R Anderson, William H Hadley, Mary A Mark, et al. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8(1):30–43, 1997.

[81] Rafael del Vado Vírseda, Pablo Fernández, Salvador Muñoz, and Antonio Murillo. An intelligent tutoring system for interactive learning of data structures. In *Computational Science–ICCS 2009: 9th International Conference Baton Rouge, LA, USA, May 25-27, 2009 Proceedings, Part II 9*, pages 53–62. Springer, 2009.

[82] Bo Xu. Artificial intelligence teaching system and data processing method based on big data. *Complexity*, 2021: 1–11, 2021.

[83] Mohammad Aamir and Mamta Bhusry. Recommendation system: state of the art approach. *International Journal of Computer Applications*, 120(12), 2015.

[84] Erik Stolterman and James Pierce. Design tools in practice: studying the designer-tool relationship in interaction design. In *Proceedings of the designing interactive systems conference*, pages 25–28, 2012.

[85] Dennis Diefenbach, Vanessa Lopez, Kamal Singh, and Pierre Maret. Core techniques of question answering systems over knowledge bases: a survey. *Knowledge and Information systems*, 55:529–569, 2018.

[86] Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. Complex knowledge base question answering: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[87] Robin K Hill. What an algorithm is. *Philosophy & Technology*, 29:35–59, 2016.

[88] Google. Dialogflow, February 2023. URL `https://cloud.google.com/dialogflow/docs`.

[89] George Terzopoulos and Maya Satratzemi. Voice assistants and artificial intelligence in education. In *Proceedings of the 9th Balkan Conference on Informatics*, pages 1–6, 2019.

[90] Charith Perera, Mahmoud Barhamgi, and Massimo Vecchio. Envisioning tool support for designing privacy-aware internet of thing applications. *IEEE Internet of Things Magazine*, 4(1):78–83, 2021.

[91] Jaideep Vaidya, David Lorenzi, Basit Shafiq, Soon Chun, and Nazia Badar. Teaching privacy in an interactive fashion. In *Proceedings of the 2014 information security curriculum development conference*, pages 1–7, 2014.

[92] Rui Rodrigues. Interactive intelligent tools for creative processes using multimodal information. In *27th International Conference on Intelligent User Interfaces*, pages 134–137, 2022.

[93] Lorin W Anderson and Lauren A Sosniak. *Bloom's taxonomy*. Univ. Chicago Press Chicago, IL, USA, 1994.

[94] Susana Masapanta-Carrión and J Ángel Velázquez-Iturbide. A systematic review of the use of bloom's taxonomy in computer science education. In *Proceedings of the 49th acm technical symposium on computer science education*, pages 441–446, 2018.

[95] Ricardo Britto and Muhammad Usman. Bloom's taxonomy in software engineering education: A systematic mapping study. In *2015 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE, 2015.

[96] Motoei Azuma, François Coallier, and Juan Garbajosa. How to apply the bloom taxonomy to software engineering. In *Eleventh annual international workshop on software technology and engineering practice*, pages 117–122. IEEE, 2003.

[97] Eur-lex, 27 April 2016. URL `gstatic.com/generate_204`.

[98] Susan Perry and Claudia Roda. Teaching privacy by design to non-technical audiences. In *Cyber Security and Privacy Forum*, pages 120–132. Springer, 2014.

[99] Susan S Conrad. Integrating data privacy principles into product design: Teaching" privacy by design" to application developers and data scientists. *Journal of Computing Sciences in Colleges*, 38(3):132–142, 2022.