ARTNAVI: AN AI-POWERED INTERACTIVE TUTORING SYSTEM FOR PERSONALIZED ART LEARNING AND FEEDBACK

Chen Cheng [1] , Moddwyn Andaya [2]

[1] Woodbridge High School, 2 Meadowbrook, Irvine, CA 92604
[2] California State University, Sacramento, 6000 Jed Smith Dr, Sacramento, CA 95819

ABSTRACT

Many students struggle to remain motivated when practicing art independently due to limited access to personalized guidance and feedback. This paper presents ArtNavi, an AI-powered art tutoring application designed to provide step-by-step instruction, real-time feedback, and motivational support. The system integrates Unity and C# with OpenAI services, combining a modular manager architecture, conversational AI, and persistent project storage to create an interactive learning environment [7]. Key implementation challenges included maintaining responsive asynchronous communication, ensuring reliable data persistence, and preserving consistent AI feedback quality; these were addressed through structured request handling, modular data serialization, and guarded message pipelines. Experimental evaluation examined feedback effectiveness across beginner and intermediate scenarios. Results showed strong performance for novice guidance, with slightly reduced effectiveness for more advanced prompts due to increased critique complexity. Overall, ArtNavi demonstrates that AI-driven tutoring can make creative learning more accessible, scalable, and supportive for diverse learners.

KEYWORDS

AI Tutoring, Art Education, Interactive Learning, Creative Feedback

1. INTRODUCTION

Art education plays an important role in supporting creativity, confidence, and academic engagement, yet many students struggle to maintain consistent practice when learning independently. Without structured feedback and encouragement, beginners often experience uncertainty about their progress and may lose motivation over time. Research has shown that participation in the arts is positively associated with improved academic and nonacademic outcomes, including higher motivation and life satisfaction among students [1]. Despite these benefits, access to personalized art instruction remains uneven, particularly for learners who cannot afford private lessons or who lack supportive learning environments.

Traditional art education models, such as classroom instruction or paid workshops, present several limitations. Classes are frequently expensive, limited in availability, or unable to provide individualized attention due to large student-to-teacher ratios. These barriers disproportionately affect students from low-income households and underrepresented communities. Studies have also indicated that arts-based programs help strengthen student engagement and self-efficacy, which are key factors in preventing academic disengagement and dropout risk [2]. When students

David C. Wyld et al. (Eds): CSITEC, NLPI, CMCA, NECOM, DMML, SP, ADCO – 2026
pp. 11-19, 2026. CS & IT - CSCP 2026 DOI: 10.5121/csit.2026.160802

do not receive sufficient encouragement or feedback, they are more likely to abandon creative practice altogether.
The long-term impact of limited access to guided art learning is significant. Students highly involved in the arts are four times more likely to be recognized for academic achievement, and low-income students with strong arts participation are more than twice as likely to graduate from college compared to their peers [3]. These findings highlight the importance of providing accessible, motivating art support systems. Therefore, addressing the gap in personalized, affordable art guidance is essential for making creative education more equitable and sustainable for diverse learners.

Methodology A demonstrated that intelligent tutoring systems can approach the effectiveness of human tutors by providing structured, step-by-step feedback. However, these systems are typically rigid and domain-specific, limiting their usefulness in creative fields. ArtNavi improves on this by offering flexible, conversational guidance suitable for open-ended art learning.

Methodology B showed that chatbots can enhance accessibility and user engagement in educational contexts. Despite this, many chatbot systems provide generic responses and lack multimodal learning support. ArtNavi addresses these gaps by combining conversational AI with image-based feedback and structured project guidance [8].

Methodology C emphasized the broader promise of artificial intelligence in education, particularly in personalization and equity. However, many proposed systems remain conceptual. ArtNavi advances this work by delivering a concrete, deployable application focused specifically on personalized creative skill development.

The proposed solution is an AI-powered art tutoring application, ArtNavi, that delivers personalized, step-by-step creative guidance using Unity, C#, and OpenAI models [9].

ArtNavi addresses the lack of accessible and motivating art instruction by providing users with continuous, individualized support that mimics the experience of working with a private tutor. When users begin a project, the system generates a structured sequence of drawing steps tailored to the user's input. Throughout the process, users can ask questions, upload sketches, and receive real-time feedback and encouragement. Because all AI requests are handled asynchronously, the application remains responsive and suitable for extended practice sessions. This design allows learners to practice independently while still receiving meaningful guidance, reducing the frustration and uncertainty that often lead to disengagement.

This approach is expected to be effective because it combines adaptive feedback, motivational reinforcement, and flexible self-paced learning within a single unified platform [10]. Many existing art learning tools rely primarily on static tutorials or prerecorded videos that cannot respond to individual student needs. In contrast, ArtNavi dynamically adjusts its guidance based on user interaction and progress, creating a more personalized learning experience. Additionally, the modular Unity architecture enables the system to remain scalable and maintainable as new learning features are introduced. For these reasons, the proposed method offers a more accessible, interactive, and supportive alternative to traditional art learning applications.
The experiment evaluated whether ArtNavi's AI feedback maintains consistent usefulness across different user skill levels. The primary goal was to identify whether beginner and intermediate users receive equally clear and actionable guidance. To test this, twenty drawing prompts were prepared and divided into beginner and intermediate categories. For each prompt, ArtNavi generated feedback that was evaluated by three independent raters using a standardized rubric measuring helpfulness, clarity, specificity, and encouragement.

The results showed that beginner prompts achieved higher average scores than intermediate prompts. This indicates that the system currently performs best when delivering foundational guidance rather than advanced critique. The most significant factor influencing performance appears to be prompt complexity, as intermediate tasks require more nuanced artistic evaluation. These findings suggest that future improvements should focus on enhancing the AI's ability to provide deeper, skill-sensitive feedback for more advanced learners.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Ensuring Reliable and Age-Appropriate AI Feedback

A major component of ArtNavi is the AI feedback system that provides guidance and encouragement to users. One potential challenge is ensuring that AI responses remain accurate, relevant, and age-appropriate for a wide range of learners. Inconsistent or overly complex feedback could confuse users and reduce trust in the system. To address this, the system could implement structured prompt templates and response validation checks to maintain consistency. Additionally, guardrails could be applied to filter unsafe or irrelevant outputs. Regular evaluation using sample user scenarios could further help ensure that the AI maintains reliable instructional quality.

2.2. Managing Asynchronous AI Communication Reliability

Another important component is the asynchronous communication between the application and the AI services. Because AI requests may take variable amounts of time, the application could experience lag or user frustration if not handled properly. Network interruptions or repeated requests may also lead to unstable behavior. To mitigate this, the system could implement request flags, timeout handling, and queue management to prevent overlapping calls. Visual loading indicators could also be used to inform users that processing is underway. These measures would help maintain a smooth and responsive user experience even when external AI services introduce delays.

2.3. Maintaining Data Integrity for User Projects

ArtNavi relies heavily on saving user progress, chat history, and images, making data integrity a critical concern. Potential issues include file corruption, incomplete savings during unexpected shutdowns, or storage inefficiencies caused by duplicate media. Such problems could lead to user frustration or data loss. To address these risks, the system could implement atomic saving operations and verification checks after each write. Duplicate detection mechanisms could also be used to prevent redundant image storage. Furthermore, a structured project folder hierarchy combined with periodic integrity validation could help ensure that user projects remain consistent, recoverable, and efficiently stored across sessions.

3. SOLUTION

ArtNavi is structured as a modular Unity application that integrates three primary components: the User Interface layer, the Core Manager system, and the AI Feedback subsystem. These components work together to provide an interactive and responsive art tutoring experience while maintaining system stability and scalability. The application was developed using Unity and C#,

with OpenAI services (ChatGPT and DALL·E) providing intelligent feedback and image generation capabilities.

The program flow begins when the user creates or opens a project through the interface shown in the system diagram. The User Interface layer handles all direct user interactions, including entering project details, selecting keywords, uploading references, and communicating through the chat panel. Once the user initiates an action, such as generating steps or requesting feedback, the request is routed to the Core Manager system. This layer, composed of scripts such as AIManager, ChatManager, and StepsManager, coordinates application logic, manages state, and prepares data for processing.

The AI Feedback subsystem then processes user input by sending asynchronous requests to the OpenAI APIs [11]. ChatGPT generates textual guidance and encouragement, while DALL·E produces reference imagery when required. After responses are received, the Core Managers update the UI with new steps, feedback, or generated content. Finally, the SavingManager persists project data, including chat history and images, ensuring continuity across sessions. This structured pipeline enables ArtNavi to deliver responsive, personalized art guidance while remaining modular and maintainable.

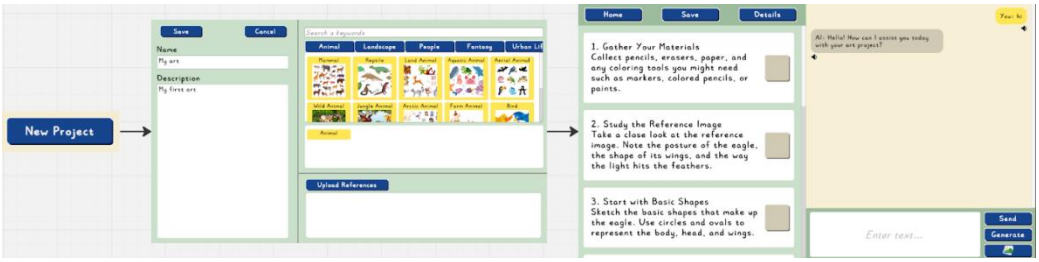

Figure 1. Overview of the solution

The AIManager component is responsible for handling all communication between ArtNavi and OpenAI services. It uses the OpenAI API to generate text feedback and guidance. This component relies on asynchronous networking and natural language processing (NLP) to produce responsive, context-aware tutoring within the application [12].

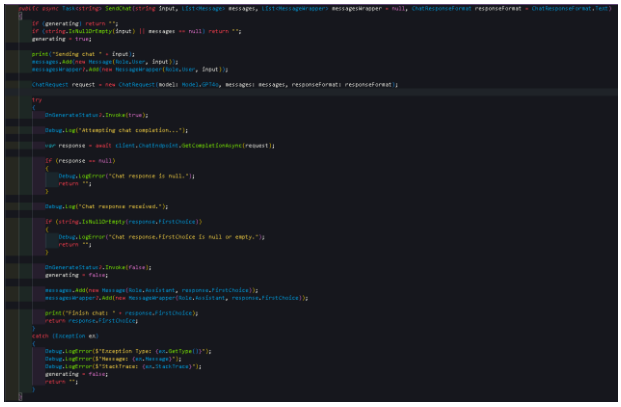

Figure 2. Implementation of the AIManager Send Chat method handling asynchronous communication with the OpenAI API.

The SendChat method is executed whenever the user submits a message in the ArtNavi chat interface. Its primary role is to forward user input to the OpenAI service and return the AI-

generated response. The method first checks the generating flag to prevent overlapping requests and validates the input. It then appends the user message to the conversation history stored in the messages list. A ChatRequest object is constructed using the GPT-4o model and the accumulated conversation context [13].

The method asynchronously calls the OpenAI Chat endpoint, allowing the application to remain responsive during processing. Once a response is received, the generating flag is cleared and the assistant's reply is appended to both the runtime message list and the optional wrapper list used for persistence. Finally, the AI's text response is returned to the caller, which then updates the user interface with the generated feedback.

The Saving Manager component is responsible for persisting all user project data to local storage. It uses JSON serialization through Newton soft.Json and standard file I/O operations. This component relies on structured data serialization to maintain project integrity and enable reliable restoration of user progress across sessions.

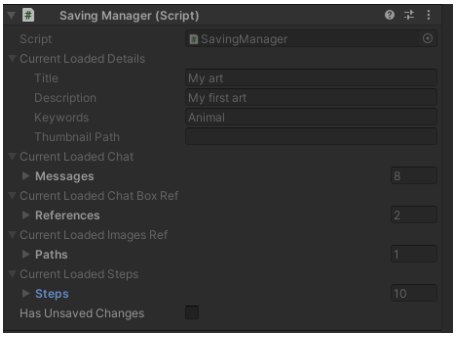

Figure 3. User interface view of the ArtNavi project workspace during drawing guidance

```
public async Task SaveChatData()

    foreach (var stepsElement in steps)
    {
        stepsData.steps.Add(new StepElementData(stepsElement.markedComplete, stepsElement.stepNumber, stepsElement.detail.title, stepsElement.detail.description));
    }

    string detailsJSON = JsonConvert.SerializeObject(details, Formatting.Indented);
    string detailsPath = Path.Combine(projectFolder, "detailsData.json");
    File.WriteAllText(detailsPath, detailsJSON);

    string chatJSON = JsonConvert.SerializeObject(chatData, Formatting.Indented);
    File.WriteAllText(chatPath, chatJSON);

    string refChatboxJSON = JsonConvert.SerializeObject(refChatboxData, Formatting.Indented);
    string refChatboxPath = Path.Combine(projectFolder, "chatboxRefData.json");
    File.WriteAllText(refChatboxPath, refChatboxJSON);

    string refImageJSON = JsonConvert.SerializeObject(refImagesData, Formatting.Indented);
    string refImagePath = Path.Combine(projectFolder, "refImageData.json");
    File.WriteAllText(refImagePath, refImageJSON);

    string stepsJSON = JsonConvert.SerializeObject(stepsData, Formatting.Indented);
    string stepsPath = Path.Combine(projectFolder, "stepsData.json");
    File.WriteAllText(stepsPath, stepsJSON);


    print($"Saved project details on: {detailsPath}");
    print($"Saved chat data on: {chatPath}");
    print($"Saved chatbox references data on: {refChatboxPath}");
    print($"Saved steps data on: {stepsPath}");

    MarkSaved();
}
```

Figure 4. Data serialization and project persistence workflow implemented in the SavingManager component

The shown code segment executes during the project save process in ArtNavi. Its purpose is to serialize runtime data structures and persist them as structured files within the project directory. The method begins by iterating through the steps list and converting each StepsElement into a StepElementData object, which is then appended to the stepsData container. This prepares step progress for serialization.

Next, multiple project components—including project details, chat history, chatbox references, reference images, and step data—are individually serialized into formatted JSON using Json Convert. Serialize Object. Each serialized string is written to a dedicated file path using File. Write All Text, ensuring separation of concerns and easier project reconstruction.

Finally, MarkSaved() is called to update the system state, indicating that no unsaved changes remain. This process ensures consistent, recoverable project persistence while maintaining modular data organization.

The ChatManager component handles real-time communication between the user interface and the AI subsystem. It manages message flow, chat bubble updates, and user interaction. This component relies on asynchronous programming and natural language processing via the AIManager to deliver responsive, conversational tutoring within ArtNavi.

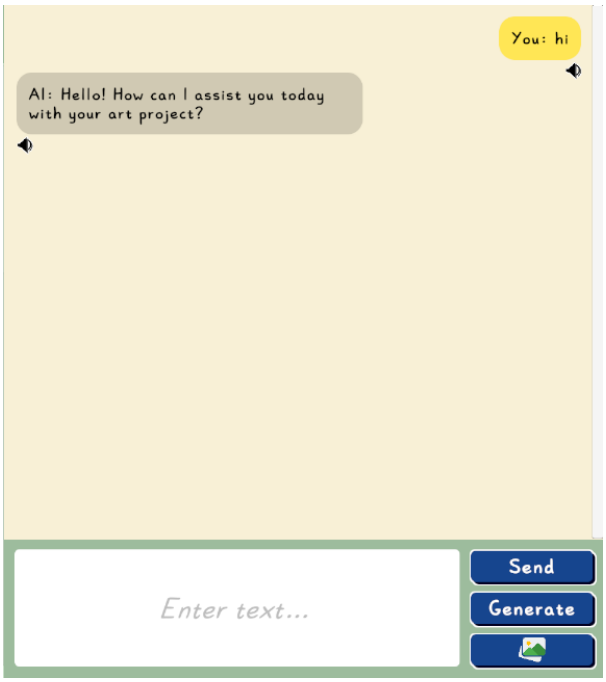

Figure 5. Chat interaction interface between the user and the AI tutoring system in ArtNavi

```
public async Task SendChatMessage(string message, bool hideUserMessage = false)   // User sending a message the getting a response back
{
    savingManager.hasUnsavedChanges = true;

    if (hideUserMessage == false)
        chatBoxManager.AddMessage(ChatBoxManager.MessageSender.User, message: "You: " + message);

    ChatBoxBubble userBubble = null;
    ChatBoxBubble aiBubble = null;

    chatBoxManager.AddMessage(ChatBoxManager.MessageSender.AI, message: "Thinking..."
                              , OnAddBubble: (ai) => aiBubble = ai
                              , OnAddOppositeBubble: (user) => userBubble = user);
    string response = await ai.SendChat(message, messages, messagesWrapper: messagesView);  // The user AND ai will add a message to the messages list after this code
    if (hideUserMessage == false) chatboxMessagesRef.Add(new ChatboxRef(ChatBoxManager.MessageSender.User, messages.Count - 2));
    chatboxMessagesRef.Add(new ChatboxRef(ChatBoxManager.MessageSender.AI, messages.Count - 1));

    aiBubble?.SetText("AI: " + response);
    userBubble?.SetText("AI: " + response);

    StartCoroutine(chatBoxManager?.UpdateUILayouts());
}
```

Figure 6. Message handling pipeline implemented in the ChatManager component

The SendChatMessage method executes whenever the user submits a message through the ArtNavi chat interface. Its role is to coordinate the full message pipeline between the UI,

AIManager, and saving system. The method begins by marking the SavingManager state as having unsaved changes. If the user message is visible, the ChatBoxManager immediately displays the user's message in the chat UI. The system then creates placeholder chat bubbles and displays a temporary "Thinking…" message while awaiting the AI response.

Next, the method asynchronously calls ai.SendChat, which sends the conversation context to the OpenAI service and returns generated feedback. Once the response is received, the method records message references for persistence and updates the AI chat bubble with the returned text [14]. Finally, the UI layout is refreshed using UpdateUILayouts to ensure proper visual alignment. This process enables smooth, real-time conversational interaction within the application.

4. EXPERIMENT

A potential blind spot in ArtNavi is whether AI-generated feedback remains consistently helpful across different user skill levels. This is critical because mismatched guidance can reduce learner motivation and engagement.

To evaluate feedback quality, an experiment was conducted comparing beginner and intermediate user scenarios. Twenty drawing prompts were prepared: ten targeting beginners and ten targeting intermediate users while maintaining similar artistic themes. For each prompt, ArtNavi generated a feedback response using the AI system. Three independent evaluators scored each response on a 1–5 scale across four criteria: helpfulness, clarity, specificity, and encouragement. The final score per prompt was computed by averaging across criteria and raters. This controlled design isolates whether user skill level affects perceived feedback quality while keeping subject matter consistent between groups.

Prompt #	Beginner Score	Intermediate Score
1	4.3	3.6
2	4.1	3.4
3	4.5	3.8
4	3.9	3.2
5	4.2	3.5
6	4.0	3.1
7	4.4	3.7
8	4.1	3.3
9	3.8	3.0
10	4.6	3.9

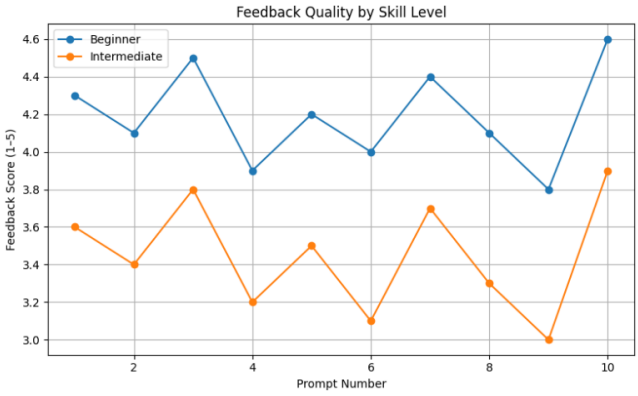

Figure 7. Comparison of AI feedback quality scores between beginner and intermediate drawing prompts

The beginner group produced a mean score of 4.19 and a median of 4.15, while the intermediate group produced a mean of 3.45 and a median of 3.45. The lowest recorded value was 3.00 (intermediate), and the highest was 4.60 (beginner). Overall, ArtNavi demonstrated strong performance for beginner-level guidance, consistently delivering clear and encouraging feedback. However, performance was moderately lower for intermediate prompts. One notable observation is the persistent gap between the two groups, suggesting that the system currently performs best when providing foundational instruction rather than more nuanced critique. This outcome is likely influenced by increased prompt complexity at the intermediate level, which requires more detailed artistic judgment. The largest factor affecting results appears to be feedback specificity, indicating that future improvements should focus on enhancing the AI's ability to deliver deeper, skill-sensitive critique for more advanced users.

5. RELATED WORK

VanLehn examined the effectiveness of intelligent tutoring systems (ITS) compared to human tutoring and traditional instruction methods. The study found that well-designed ITS platforms can approach the effectiveness of human tutors by providing step-by-step guidance and immediate feedback to learners [4]. While this demonstrates the potential of AI-driven instruction, most ITS implementations are highly domain-specific and structured around fixed problem sets. They often lack flexibility, conversational depth, and support for creative disciplines such as art. ArtNavi improves upon this approach by integrating conversational AI and multimodal feedback, enabling more adaptive, open-ended guidance suitable for creative learning environments.

Følstad and Brandtzæg explored the use of chatbots in human-computer interaction, highlighting their growing role in educational support and user engagement. Their work shows that conversational agents can improve accessibility and provide on-demand assistance to users [5]. However, many chatbot systems remain limited in contextual understanding and personalization, often delivering generic responses that reduce instructional effectiveness. Additionally, most implementations focus primarily on text interaction without incorporating visual learning support. ArtNavi addresses these limitations by combining conversational AI with image-based feedback and structured step guidance, creating a more personalized and interactive tutoring experience for art learners.

Holmes, Bialik, and Fadel discussed the broader role of artificial intelligence in education, emphasizing its potential to personalize learning and expand access to instructional support [6]. Their analysis highlights that AI systems can adapt to individual learner needs and help reduce educational inequities. However, the work also notes that many AI education tools remain conceptual or lack practical classroom-ready implementations. Furthermore, few systems specifically target creative skill development. ArtNavi builds upon these insights by delivering a concrete, deployable application that applies AI personalization directly to art education, offering real-time feedback, project guidance, and motivational support within a unified learning platform.

6. CONCLUSIONS

Although ArtNavi demonstrates strong potential as an AI-assisted art learning platform, several limitations remain. First, the current system relies heavily on external AI services, which may introduce latency, usage costs, and dependency risks. Future work could implement response caching, request batching, or hybrid local models to reduce reliance on remote services. Second, while the application provides personalized guidance, its progress recognition and skill assessment capabilities are still limited. With more time, computer vision models could be integrated to automatically evaluate drawing quality and provide more objective feedback. Third,

the user interface, while functional, could benefit from further usability testing across different age groups to improve accessibility and clarity. Additional improvements could include user accounts, cloud synchronization, and collaborative features. Addressing these areas would strengthen scalability, robustness, and educational effectiveness.

ArtNavi demonstrates how AI-driven systems can make creative learning more accessible, personalized, and engaging [15]. By combining conversational guidance, structured project support, and persistent data management, the platform provides a strong foundation for future AI-assisted art education tools aimed at supporting diverse learners.

REFERENCES

[1] Martin, Andrew J., et al. "The role of arts participation in students' academic and nonacademic outcomes: A longitudinal study of school, home, and community factors." Journal of Educational psychology 105.3 (2013): 709.

[2] Charmaraman, Linda, and Georgia Hall. "School dropout prevention: What arts-based community and out-of-school-time programs can contribute." New Directions for Youth Development 2011.S1 (2011): 9-27.

[3] Mikhalchenkova, Natalya Alekseevna, et al. "Designing a navigator and developing a classification of methods of art education and cultural studies." EurAsian Journal of Biosciences 14.2 (2020).

[4] VanLehn, Kurt. "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems." Educational psychologist 46.4 (2011): 197-221.

[5] Følstad, Asbjørn, and Petter Bae Brandtzæg. "Chatbots and the new world of HCI." interactions 24.4 (2017): 38-42.

[6] Holmes, Wayne, Maya Bialik, and Charles Fadel. Artificial intelligence in education promises and implications for teaching and learning. Center for Curriculum Redesign, 2019.

[7] Kulkarni, Pradnya, et al. "Conversational AI: An overview of methodologies, applications & future scope." 2019 5th International conference on computing, communication, control and automation (ICCUBEA). IEEE, 2019.

[8] Vo, Philippe QN, et al. "Image-based feedback and analysis system for digital microfluidics." Lab on a Chip 17.20 (2017): 3437-3446..

[9] Roumeliotis, Konstantinos I., and Nikolaos D. Tselikas. "Chatgpt and open-ai models: A preliminary review." Future Internet 15.6 (2023): 192.

[10] Ham, MyungJoo, and Geunsik Lim. "Making configurable and unified platform, ready for broader future devices." 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). IEEE, 2019.

[11] Auger, Tom, and Emma Saroyan. "Overview of the openaiapis." Generative AI for Web Development: Building Web Applications Powered by OpenAI APIs and Next. js. Berkeley, CA: Apress, 2024. 87-116

[12] Nadkarni, Prakash M., Lucila Ohno-Machado, and Wendy W. Chapman. "Natural language processing: an introduction." Journal of the American Medical Informatics Association 18.5 (2011): 544-551.

[13] Wu, Yiqi, et al. "Gpt-4o: Visual perception performance of multimodal large language models in piglet activity understanding." arXiv preprint arXiv:2406.09781 (2024).

[14] Sanda, Emanuel. "Artificial Intelligence Algorithms and the Facebook Bubble." Proceedings of the International Conference on Economics and Social Sciences. 2022.

[15] Da Fonseca, Ana Taveira, Elsa Vaz de Sequeira, and Luís Barreto Xavier. "Liability for AI Driven Systems." Multidisciplinary perspectives on artificial intelligence and the law. Cham: Springer International Publishing, 2023. 299-317.