Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #47395

BOARD # 63: AI Chatbot for Enhancing Troubleshooting in Engineering Labs

Marshall Ismail, Worcester Polytechnic Institute

This paper was co-authored by three Worcester Polytechnic Institute undergraduate students pursuing degrees in Mechanical Engineering and Robotics Engineering. Having previously completed the mechanical engineering laboratory course of which the study is based on, they are familiar with the challenges that students often face when working on lab assignments without sufficient guidance. Their collective experience in the course and guidance by Professor Sabuncu inspired them to create a AI-based chatbot aimed at providing targeted support for students, helping them navigate complex lab assignments with ease.

Devin Kachadoorian, Worcester Polytechnic Institute
Sahil Mirani, Worcester Polytechnic Institute
Dr. D. Matthew Boyer, Clemson University

Dr. Boyer is a Research Associate Professor in the Department of Engineering and Science Education and an Educational Proposal Writer in the College of Engineering, Computing and Applied Sciences.

Mr. Tim Ransom, Clemson University

Computer Science Lecturer, PhD candidate with the Engineering and Science Education Department at Clemson University.

Prof. Ahmet Can Sabuncu, Worcester Polytechnic Institute

Dr. Sabuncu holds a Ph. D. in Aerospace Engineering from Old Dominion University. Dr. Sabuncu's professional interests spans from engineering education research, history of science and engineering, thermo-fluids engineering, and microfluidic technology.

AI Chatbot for Enhancing Troubleshooting in Engineering Labs

Introduction

Engineering education fosters critical thinking, creativity, and professional identity through hands-on laboratory experiences that bridge theoretical knowledge and real-world application. Laboratories develop practical engineering skills and cultivate problem-solving abilities, sensory awareness, and technical intuition, preparing students to tackle complex, real-world challenges with confidence and adaptability [1]. In these settings, students frequently turn to teaching assistants (TAs) for assistance with lab procedures, equipment setup, and troubleshooting. This dynamic creates a dependency that, while helpful in the moment, can lead to challenges for both students and TAs. The repetitive nature of these inquiries significantly burdens TAs, who usually cannot answer everyone's questions throughout the laboratory class times.

Furthermore, certain student questions need consistent answers that the lead instructor proves correct. Another challenge is establishing a structured support diagnostic meant to answer student problems in a way that guides students to their answers rather than revealing them immediately. This allows students to engage in classroom learning as they are incentivized to find the answer using materials and the structured support provided by technology. Thus, having a techhat is built off of resources meant to push students to learn actively is beneficial for a class that requires constant engineering troubleshooting to succeed.

In response to these challenges, this study explores the integration of artificial intelligence (AI) as an innovative tool to support laboratory-based mechanical engineering education. Specifically, we developed an AI chatbot designed to provide immediate, on-demand assistance in engineering experimentation classes. These courses instruct students on using engineering concepts in laboratory experimentation through project experimentation using Arduino or Raspberry Pi to control circuits. Each project requires students to use a variety of sensors, motors, and other mechanical hardware to measure certain attributes or complete certain tasks. Students must be able to gather the correct data and show their project is fully functional to pass the assignment. The chatbot aims to address common student questions, streamline instructional support for students, and foster a deeper understanding of these engineering concepts.

This study's primary purpose is to evaluate this AI chatbot's effectiveness in enhancing the educational experience in engineering labs. The chatbot is expected to offer consistent, high-quality support while encouraging students to engage in critical thinking and independent problem-solving, and it employs advanced problem-solving frameworks, such as issue trees and first principles, as well as Socratic questioning, to guide students through complex challenges. The AI chatbot aims to build student confidence and competence in navigating engineering problems through these methods.

To assess the chatbot's impact, we use the following research questions:

- How effectively is the AI chatbot improving students' understanding of lab procedures and instrumentation?

- In what capacity does the chatbot promote deeper conceptual thinking and independent problem-solving?

These questions form the foundation of a mixed-methods evaluation approach, which combines qualitative student feedback, surveys, and an analysis of the chatlog. By addressing the troubleshooting issues in engineering education with an AI-driven solution, this study analyzes using an AI chatbot in an academic setting to enhance student learning. This chatbot can be a scalable tool for other engineering laboratory classes, providing consistent, accessible, and innovative support to students while alleviating the demands placed on instructors and TAs.

Literature Review

This literature review examines four key themes central to understanding the role of AI chatbots in educational contexts. First, the application of AI chatbots in educational support is discussed, emphasizing their potential to foster independent learning and critical thinking. Second, the roles and challenges of TAs in engineering labs are analyzed, highlighting the demand for supplementary technological tools. Third, problem-solving frameworks, such as first principles and issue trees, are explored for their relevance in engineering education and integration into chatbot design. Finally, assessment methods for evaluating educational technology interventions are reviewed to ensure the effectiveness and scalability of these tools. These themes collectively provide a foundation for exploring how AI chatbots can enhance learning outcomes and address critical challenges in engineering education.

AI Chatbots in Educational Support

AI chatbots are increasingly recognized as valuable tools in educational environments, offering scalable, personalized support for learners across disciplines. These conversational agents use natural language processing (NLP) to engage students in interactive learning, assisting with tasks such as answering questions, providing feedback, and encouraging independent problem-solving. Their integration into educational settings addresses critical gaps in instruction, particularly in large-scale classrooms with limited one-on-one support.

The ability of AI chatbots to provide immediate, on-demand assistance has demonstrated significant potential in improving student engagement and learning outcomes. Chatbots help bridge the gap between students and instructors by enabling real-time interaction, offering consistent responses, and fostering a supportive learning environment [2, 3]. This capability is particularly relevant in engineering education, where students often require timely guidance during complex problem-solving tasks [4].

The deployment of chatbots in education is not without challenges. Usability issues, such as unclear communication or difficulty handling nuanced student queries, can limit their effectiveness. Addressing these limitations requires iterative design processes and user feedback to optimize chatbot functionality. Despite these challenges, the ongoing evolution of AI chatbots underscores their potential to revolutionize educational support systems, particularly in STEM-focused domains [5].

Teaching Assistant Role and Challenges in Engineering Labs

Teaching assistants (TAs) play a critical role in engineering education by providing students individualized guidance and support during laboratory sessions. These responsibilities often include answering technical questions, assisting with experimental setups, and troubleshooting complex problems. However, the high dependency on TAs in large-scale classes frequently leads to challenges, including limited availability, inconsistent instructional quality, and an overwhelming workload for TAs. These issues can negatively affect the learning experience for students and impede the development of critical problem-solving skills.

In engineering labs, the demand for TA support intensifies during practical sessions, where students encounter unforeseen errors or require clarification on intricate concepts. This demand often exceeds the capacity of available TAs, leading to delays in addressing student concerns and creating a bottleneck in the learning process. Additionally, the repetitive nature of many inquiries can detract from TAs' ability to provide meaningful, in-depth guidance [6].

AI chatbots offer a scalable solution to mitigate these challenges by complementing the role of TAs. These tools can handle repetitive, procedural queries, allowing TAs to focus on more complex instructional tasks. Moreover, AI systems equipped with problem-solving frameworks can promote response consistency, reducing variability in student support [7]. Despite these advantages, integrating AI tools must address ethical considerations and potential limitations, such as ensuring that students continue to develop interpersonal and teamwork skills essential for engineering practice [8]. TAs and AI chatbots can create a more efficient and effective learning environment for engineering students.

Problem-Solving Frameworks in Engineering Education

Problem-solving is a foundational skill in engineering education, where students must analyze complex scenarios and devise practical solutions. Engineering curricula often incorporate structured frameworks such as first principles reasoning and issue trees to effectively develop these skills. These frameworks provide systematic approaches for deconstructing problems into manageable components, encouraging a deeper understanding of concepts and fostering critical thinking.

First principles reasoning emphasizes breaking a problem down to its most fundamental elements, enabling students to build solutions from the ground up. This approach nurtures innovative thinking and enhances the ability to tackle unfamiliar problems. Issue trees, on the other hand, facilitate a hierarchical breakdown of complex challenges, guiding learners through the systematic exploration of potential solutions. These frameworks are useful in engineering labs, where troubleshooting and iterative problem-solving are integral to learning [9].

AI chatbots have demonstrated significant potential in embedding these problem-solving frameworks into student interactions. AI systems can guide students through structured inquiry by employing first principles of reasoning and issue trees, prompting them to think critically and explore alternative approaches. This aligns with educational goals prioritizing finding the correct answer and understanding the underlying processes [10].

However, challenges remain in ensuring students fully engage with these frameworks rather than relying solely on AI-generated guidance. Effective integration requires careful design to balance automated support with opportunities for independent learning and cognitive development [11]. Through these frameworks, AI tools can play a pivotal role in equipping engineering students with robust problem-solving skills essential for their professional success.

Assessment Methods for Educational Technology Interventions

Evaluating the effectiveness of educational technology interventions is essential to ensure their impact on learning outcomes and to guide iterative improvements. In the context of AI chatbots in engineering education, robust assessment methods are necessary to measure their contributions to student engagement, problem-solving abilities, and overall learning experiences. These evaluations require quantitative and qualitative approaches to capture the nuanced interactions between students and technology.

Quantitative measures often include improvement in test scores, task completion rates, and reductions in TA workloads. These indicators provide objective insights into the performance and efficiency of the chatbot in supporting instructional goals. Qualitative methods, such as student surveys, focus groups, and analysis of chatbot-student dialogues, complement quantitative data by exploring user experiences, perceived value, and areas for enhancement [12].

Mixed-methods approaches integrate quantitative and qualitative data and are increasingly favored for their comprehensive perspective. For instance, analyzing chatlogs can reveal patterns in student inquiries, the chatbot's ability to foster critical thinking, and how effectively it employs problem-solving frameworks like first principles reasoning. Longitudinal studies are particularly valuable for assessing the sustained impact of these interventions on student learning and their alignment with educational objectives [13].

However, challenges in assessment persist, including the ethical considerations of data privacy and the potential biases in interpreting user feedback. Addressing these issues requires transparent methodologies and a commitment to refining the design of AI-driven educational tools based on evidence-based practices [14]. Through rigorous assessment, AI chatbots can be optimized as transformative tools in engineering education.

AI Chatbot

As mentioned, a chatbot is a chat-based algorithm that uses natural language processing (NLP) algorithms to converse with the user. OpenAI's ChatGPT is an example of a chatbot because it uses both natural language processing and proprietary algorithms to communicate with users in a conversation-like manner. The algorithm analyzes the text, and then associated data with the subject of the text is returned to the user whether the user asks a question or makes a statement. For this research study, a ChatGPT-based language model was trained and used to implement a troubleshooting framework that guided students in their engineering laboratory class. This model was developed in Flowise, an open-source chatbot builder, using specific design changes such as language model integration, embeddings, vector databases, document stores, retriever tools, and moderation tools. The chatbot uses the OpenAI API to obtain access to the ChatGPT model 4o mini, and it is designed using Flowise to allow for future changes to the specific design elements mentioned above to be easily made, as Flowise uses pre-programmed elements rather than hard code to make changes to the chatbot's functionality.

Methodology

The study investigates the practicality and usefulness of the chatbot for learning engineering concepts, and this study aims to improve this chatbot learning capability through an understanding of user feedback, error identification, and general improvement in guidance capabilities. By addressing errors, enhancing guidance capabilities, and refining its learning mechanisms, the research team is determined to make the chatbot a valuable educational resource. The methodology revolves around structured objectives, detailed procedures, and iterative improvements based on real-world classroom interactions. The initial Flowise framework of the chatbot and the various other outside tools, such as Pinecone and Postgres, are explored in detail in this section. The chatbot evolved through various developmental stages, through testing and analyzing student engagement through surveys and chatlog data. Finally, the methodology limitations are discussed to guide future improvements.

Objectives

The study addresses several key questions to understand the feasibility and effectiveness of an AI chatbot in an educational setting. This exploration involves determining how engineering troubleshooting techniques, including Socratic questioning, first principles, and decision trees, can be adapted to create a robust chatbot framework. The research seeks to uncover how students utilize the chatbot, its value in resolving complex engineering problems, and its potential for scalability across different courses. The questions below guide its development and refinement:

1. In what capacity can Socratic questioning and other engineering troubleshooting techniques (such as decision trees) be used to develop a chatbot to assist students in engineering classes?
2. In what ways do students use the AI chatbot in a classroom setting?
3. How practical and useful is an AI chatbot when used as an engineering tool in a classroom setting, and can future implementations be further developed?
    a. How can it be used in the same course?
    b. How can it be expanded upon to be used in different courses?

Development and Testing of Troubleshooting Frameworks

The first step in improving the chatbot's functionality involves creating and implementing a robust troubleshooting framework. The primary objective of this framework is to enhance the chatbot's ability to assist students in resolving issues encountered during lab work. Various troubleshooting strategies were developed based on effective teaching of engineering applications, including Socratic questioning, critical thinking techniques, hands-on applications, and trial-and-error problem-solving.

These troubleshooting strategies are referenced in the chatbot's pdf library through the research documents exploring how industry voices perceive troubleshooting, its importance in developing a troubleshooting mindset in student learning, and how Socratic questioning, in particular, can be used to acquire knowledge and develop intellectual skills [15, 16]. We wrote the chatbot's instructions and meticulously changed over the testing period to yield better responses from the chatbot. Because different student questions in a course with a wide range of issues require different response structures, an explicit one-size-fits-all response design with each of the implemented strategies cannot be given to the chatbot. Therefore, the chatbot is set to deal with various problems individually without human supervision using information from the professor's laboratory documents and information available through OpenAI's pre-trained language model of ChatGPT 4o. Once the response instructions and the library are implemented, the chatbot's responses will be tested.

User Feedback Collection and Analysis

Two ways to obtain student data: surveys and chatlog data. The surveys contain the student opinions towards its use as a tool, the effectiveness of the chatbot to their learning, and the chatbot's performance. Feedback identifies shortcomings, informs enhancements to troubleshooting processes, evaluates the chatbot's utility in the class and refines the chatbot's guidance capabilities. Surveys are administered at the end of the academic term, allowing students to reflect on the overall chatbot experience. The survey was created using Qualtrics and administered using the Qualtrics participant link. Once the term ended, students received the Qualtrics participant link electronically through email. We did this using the mass email distribution tool on Qualtrics. After several days, students received another follow-up email reminding them to complete the survey to get more responses. All emails were sent on behalf of the professor's email address to make the email seem official rather than spam. Chatlog data was obtained by connecting Langfuse to Flowise. This service allows the OpenAI account owner, the professor, to view every student request to the chatbot and every response generated by the chatbot. The research team then analyzes this data manually using a coding system to understand the actual effectiveness of the chatbot responses. This coding system categorized student questions and responses into three categories: solution seeking, concept clarification, and process documentation. Solution seeking means the student is asking the chatbot for troubleshooting help or a solution to a problem occurring in their work. Concept clarification means the student uses the chatbot to explain a certain engineering concept they are unfamiliar with. Finally, process documentation means the student asks the chatbot about the specific requirements for an assignment. All chatlog interactions are permanently stored through Langfuse and can be accessed securely only through the professor's account.

Flowise Implementation and Chatbot Creation

The chatbot was developed using Flowise, an open-source low-code framework for building conversational AI systems. The implementation integrates OpenAI's large language models and employs custom chatflow guidelines and parameters to handle student queries for a mechanical engineering course. To develop a similar chatbot, we used the following essential components in the current architecture:

1. Core Components:
   - **Language Model Integration**: The current architecture uses OpenAI's API to handle all user queries. This API provides the language model backend, specifically leveraging the ChatOpenAI module for natural language processing to understand user queries and

generate appropriate responses. The model configuration includes additional parameters, such as temperature and max token, to optimize responses for pedagogical clarity.
- **Embeddings**: The OpenAI module uses the text-embedding-ada-002 model to generate vector embeddings for document retrieval tasks. These embeddings enable efficient access to information from source documents.

2. Data Management:
- **Vector Database**: The chatbot employs Pinecone to store and retrieve document embeddings. Deployed on AWS, the vector database ensures quick similarity searches to match user queries with relevant course materials. Metadata filtering further enhances this process.
- **Document Store**: Source documents are uploaded into a document store where metadata filters allow the Pinecone database to filter the documents. The documents are divided into chunks for optimal retrieval performance, with the standard configuration of 1,000 characters per chunk and a 200-character overlap.

3. Customization:
- **Retriever Tools:** Custom retriever tools are designed to filter and fetch course-specific documents based on user input. Each tool is linked to a specific lab module and microcontroller, enabling targeted query handling. These tools are integrated with Pinecone database modules using specific metadata filters to ensure precise document retrieval.
- **Moderation Element:** An optional OpenAI moderation tool ensures compliance with usage policies, enhancing system reliability.
- **Custom Tool:** A custom tool prevents the LLM from providing direct coding responses; instead, it offers pseudo-code solutions to help troubleshoot user code. Hardcoded in JSON, this tool enforces predefined system guidelines by preventing users from bypassing these restrictions.

4. Memory System:
- **Buffer Memory:** The memory element allows the chatbot to access earlier messages in the conversation, enabling more natural interactions. This functionality helps the chatbot isolate the user's issues, troubleshoot effectively, and enhance the user's understanding.

5. Interaction Flow:
- **Tool Agent:** The tool agent oversees the conversational interaction with the user. It adheres to predefined pedagogical guidelines, emphasizing analysis driven by source documents and employing Socratic questioning to promote critical thinking and problem-solving skills. The agent dynamically manages tool selection and execution in response to user queries, providing adaptive and context-aware assistance.

6. Scalability and Maintenance:
- The architecture's modular structure allows new users to easily add retriever tools and update source documents as needed.

7. Security:

- The chatbot is deployed on the institution's servers and only provides access to students to ensure privacy and safety. While the deployment is secured, only authorized members can access the chatbot's backend. However, student chatlogs remain accessible solely to the research team, safeguarding user data.

Chatbot Version Progression

First Iteration: The initial prototype introduced the fundamental components of the chatbot, including a Postgres database, a document uploader, a character text splitter, and a conversational retrieval QA chain (Figure 4). These elements provided the essential framework for retrieving and processing information. Postgres was chosen as the database management system due to its reliability in handling structured data. The overuse of the document uploader highlighted the need to transition to a document store to reduce system clutter and enhance modularity.

The conversational retrieval QA chain was sufficient in answering user queries by extracting relevant document-based information. However, a key limitation emerged—while the chatbot performed well with text-based queries, it lacked the ability to process image uploads. This limitation restricted the chatbot from fully leveraging GPT-4o's capabilities in visual problem-solving.

Second Iteration: Building on the lessons from the first iteration, the team replaced the document uploader with a document store to better manage and retrieve various file types. A major goal of this iteration was to integrate image uploads, but all of Flowise's conversational chains lacked native support for image processing. To overcome this, the team explored alternative methods and identified the "Tool Agent" as a potential solution. The Tool Agent could simulate the functionality of a conversational chain but had a big limitation, where it was unable to directly interact with the database.

To resolve this, a "Retriever Tool" was utilized, functioning both as a "Tool" to interface with the Tool Agent and as a "Retriever" to extract relevant data from the database. A custom tool was also integrated to prevent the chatbot from providing direct coding solutions, focusing instead on pseudo-code guidance to adhere to a structured troubleshooting framework. This revised architecture was deployed on the institution's servers for classroom use.

Third Iteration: While the second iteration improved the chatbot's ability to retrieve information, user testing revealed a persistent issue: the chatbot was not effectively utilizing lab documents for answering queries. Instead, it frequently defaulted to the LLM's knowledge base rather than sourcing information from the provided documents. The issue stemmed from the Retriever Tool's inability to filter and prioritize relevant information correctly.

To address this, the team refined the retrieval mechanism by developing multiple retriever tools, each configured to extract data from a specific lab document. By creating modular retrievers according to lab activities, the chatbot was able to return more precise results. Additionally, retriever tools were specialized to correspond with specific microcontrollers—such as Arduino or Raspberry Pi—allowing the chatbot to tailor responses based on the user's hardware setup. These enhancements aimed to improve the chatbot's ability to troubleshoot user-specific issues, offering a more personalized troubleshooting process. After integrating these refined retriever tools, the chatbot was redeployed, showcasing marked functionality and user assistance improvements.

Fourth Iteration: Despite the improvements made in the third iteration, inconsistencies were recorded in the Retriever Tool's ability to retrieve the most relevant documents. The root cause was identified as an inefficient retrieval process that pulled an excessive number of files at once, often leading to mismatches between user queries and retrieved documents. This problem, though reduced, persisted even after increasing the number of specialized retriever tools.

To resolve this, the team looked to implement metadata filtering, allowing the chatbot to filter documents based on predefined metadata fields such as document type, topic, and associated hardware. However, a technical limitation emerged—Postgres lacked native support for metadata filtering. To address this, the team evaluated alternative databases and ultimately selected Pinecone for its robust metadata filtering capabilities and online accessibility, which facilitated easier testing across local and server environments.

With Pinecone integrated, the chatbot could now retrieve only the most relevant documents based on metadata tags, significantly enhancing accuracy. A general retriever tool was also added to address the dual requirements of lab activities and personal projects for the mechanical engineering course. This tool extended the chatbot's utility to support personal projects by leveraging the LLM's extensive knowledge base rather than limiting the assistance to only lab activities. Once all improvements were implemented, the chatbot underwent extensive testing. The results confirmed that it could accurately retrieve source documents based on user queries while also effectively assisting with independent projects. With these final enhancements, the updated chatbot architecture was successfully deployed on the institution's servers, ensuring a more efficient and versatile tool for students.

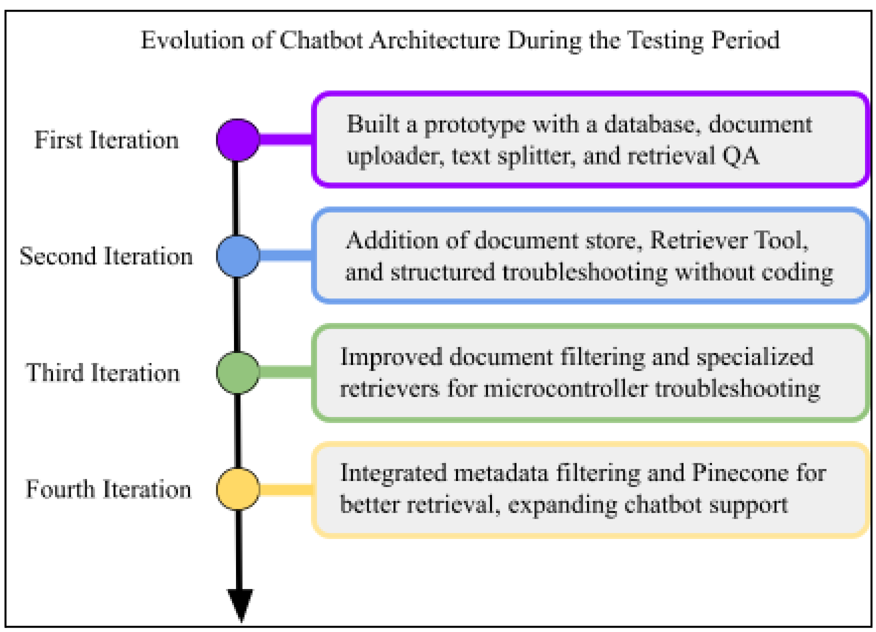

Figure 1: Evolution of Chatbot Architecture Chart

Data Collection

Participants

This study was approved by the institutional review board of the research institution, which allows this study to be conducted using human participants. The research is conducted ethically and with proper participation protection, adhering to necessary regulations set by the IRB. The chatbot and survey participants are actual students in the course instructed by the last author of this study. Students are given the option to use this AI chatbot tool at the beginning of the course, but students are not required to use the tool. Students are asked to review and sign the consent form the research team created before they start using the chatbot. This research subject document explains the goals of the research, why the students' feedback is necessary, and any benefits or drawbacks in participating. As stated in the document, students are not required to participate in the study as a part of the class and can choose not to join the research study.

Duration of Research Study

The institution is based on a quarter academic system, and academic terms are typically limited to roughly seven full weeks of classes. Specifically, the engineering lab class meets in person twice a week for three hours, although students can do some laboratory assignments at home and outside of official class times. The research team tested the chatbot for two academic terms with the first lasting from October 21 to December 13, 2024, and the second lasting from January 15 to March 7, 2025.

Limitations

While the AI chatbot shows promise as a tool to enhance engineering education, its implementation within the course came with several limitations, specifically when the chatbot was initially tested. One of the key limitations was the program, chatbot, and website errors experienced by students throughout the testing process. Because the chatbot was launched without extended testing periods, many students encountered various errors and bugs with functionality. The chatbot was initially launched for a 10-day period to have students test for bugs; after this pilot period, the iterations mentioned above were implemented. The bugs could significantly impact how students viewed the chatbot overall, even after the changes were made. Another significant limitation was the inconsistency of information regarding the different laboratory assignments. Initially, the chatbot managed to confuse pre-loaded course data. For example, if a student asked for information regarding a specific assignment, they would respond with the requirements of a different assignment. This was seen as a significant error by the research team and was the primary reason for using metadata and the Pinecone database structure. Although this was early in the research testing period and was quickly fixed, several students could have noticed this error, which negatively impacted their view of the chatbot when it gave an incorrect response.

The chatbot also demonstrated limitations in addressing highly specific engineering troubleshooting questions or scenarios that it cannot answer without having access to instrument data and results. For example, many instruments such as an Arduino, thermistors, thermocouples, strain gauges, and motors are used in laboratory experiments and without access to the data received from these components, the chatbot would be unable to give precise responses for how to troubleshoot an issue. Usually, the chatbot would ask the user for data. Still, for a more helpful chatbot, the research team would prefer the chatbot to have continuous access to the data or software output from the mechanical components. This would allow the chatbot to understand the exact issue the student has and streamline the troubleshooting process between the student and the chatbot. This would mimic how a TA oversees the data collected when helping troubleshoot the specific problems a student may have.

These issues, while expected in a developmental phase, underscore the importance of continuous testing and refinement of the chatbot, even during the live user testing. By resolving these errors and expanding the chatbot's capabilities, the research team worked to ensure the chatbot's long-term viability as a supportive tool for engineering education.

Results

Survey Results

The chatbot was tested across two academic terms with the first lasting from October 21 to December 13, 2024, and the second lasting from January 15 to March 7, 2025. During the last week of class, the Qualtrics survey was sent to the students, and 12 out of 59 total students completed the survey from the first academic term and 8 out of 60 students completed the survey from the second academic term.

In the survey, students were asked which tasks they used the chatbot for with students being able to make multiple selections. Of all survey respondents, 76% of students claimed to seek chatbot assistance with lab experiments, 65% used the chatbot for debugging code, 59% used the chatbot for concept explanations, and 35% used the chatbot for answering theoretical questions.

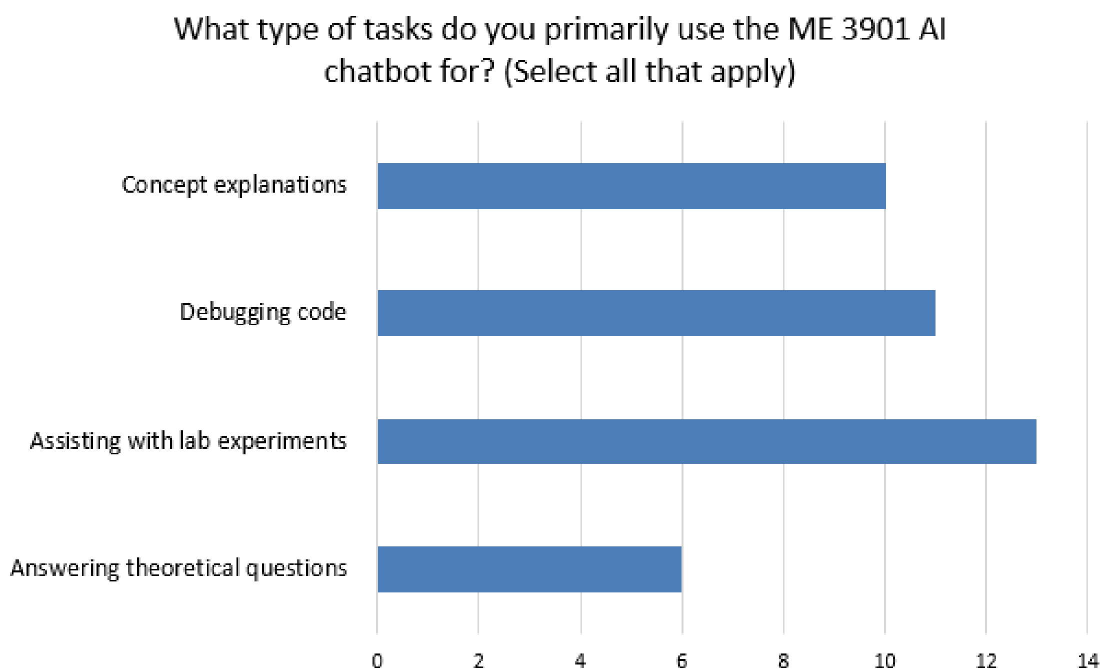

Figure 3. Ways in which students use the AI chatbot

On a scale of 1-5, the surveyed students rated average was 4.06 for the chatbot's helpfulness in assisting with understanding theoretical concepts. For solving code-related problems, the average was 3.79. For setting up and calibrating experiments, the average was 3.57. For both improving critical thinking skills and improving problem-solving skills the average was 3.00.

On a scale of 1-5 (5 being most helpful), how helpful do you find the chatbot in assisting with:	Average
Understanding theoretical concepts	4.06
Solving coding-related problems	3.79
Setting up and calibrating experiments	3.57
Improving your critical thinking skills	3.00
Improving your problem-solving skills/strategy	3.00

Table 1. Helpfulness of the chatbot across certain skills

When asked about the chatbot's ease of use, 38% of respondents rated it as "neutral," 44% rated it as "easy" to use, and 19% rated it as "very easy" to use. Students who claimed the chatbot's ease of use was neutral also responded that sometimes the chatbot did not fully understand their questions; on one occasion, a student had the chatbot respond "Hmm, I'm not sure" to all of their questions. No responses claimed that the chatbot was difficult to use. Generally, the surveyed students agreed that the chatbot should be used in educational settings or has potential to be used in educational settings after some improvements; they believed the chatbot was beneficial as a supplemental tool for quick questions. There were a few students who expressed the opposite opinion believing the chatbot to be an unethical replacement for teaching assistants. For a survey question regarding student use of the chatbot in other courses, 60% of survey respondents said they would use this chatbot in future courses, 27% said no, and 13% said maybe.

Chatlog Results

Student-chatbot discussions were recorded and analyzed to determine the chatbot's ability to assist in troubleshooting based on the developed framework, provide factual information to students, and understand the success of the chatbot's aid. The following chatlog data is taken from multiple sessions to show the diversity of questions and situations the chatbot was asked to solve.

The following chatlogs come from Lab 4 of the class which involves monitoring the change in temperature of hot water in a soda can. The students then modify the soda can to cause the water to cool off faster. A thermistor is a component in the lab which changes electrical resistance in response to temperature changes. Students use a thermistor to measure the temperature inside the soda can based on its current resistance.

Session #1:
Initial Prompt: "i am about to start lab 4 for the heat transfer experiment what hertz would i use for measuring temperature"
Discussion Description: Initially, the chatbot responds by saying that the frequency of measurements typically depends on the specific sensor and setup, so it does not give a direct answer. Instead, the chatbot explains that thermistors and thermocouples provide continuous readings instead of discrete frequencies, and the temperature would be read at intervals defined by the code. The chatbot asks for clarification on the specific sensor and setup used to help guide the user better. The student responds by saying that they are using a thermistor, and the chatbot replies that a specific frequency does not need to be set for thermistors and provides information about setting up and calibrating a voltage divider circuit for the thermistor. The student then responded again, asking which frequency to use, between 1, 10, and 0.1. The chatbot then explains the differences between the three options and the application in the experiment. A higher frequency sampling rate would mean more readings per second and should be used if rapid temperature changes. Likewise, a lower frequency sampling rate would mean fewer readings per second and are better suited for monitoring stable environments. The chatbot recommends 1 Hz to start and to increase to 10 Hz if the temperature changes happen too fast. The chatbot offers to help the student in exploring how this can be implemented into the lab setup.
Analysis: In this session, the student uses the chatbot to help in the lab experiment and to clarify concepts related to frequency. The student needed help determining the sampling rate when recording the water temperature in the soda can. The chatbot initially thought the student was asking about the frequency at which the thermistor would measure temperature, which happens continuously. This misconception prevented the chatbot from answering the student's question until further clarification. After the student clarified, the chatbot could correctly define the different frequencies, what they mean in the experiment context, and which frequency would be best. However, the chatbot failed to explain to the student that using higher frequency readings produces more noise in the data and lower frequency readings should be taken during this lab.

Session #2:
Initial Prompt: "I am working on the thermistor lab and am having trouble writing my own code"
Discussion Description: The chatbot responds with a few clarifying questions regarding the code's functionality, components used in the lab, and sections of the code the student is having trouble with. Afterward, the student asks:

"what variables do i need to create for temperature?"

The chatbot responds with variables useful to the lab, including voltage reading (to store the voltage across the thermistor), thermistor resistance (to store the calculated resistance of the thermistor), reference resistance (to store the value of the reference resistor in the voltage divider circuit), temperature (to store calculated temperature), reference voltage (to store the reference voltage which is usually 5V or 3.3V), and calibration constraints for the Steinhart-Hart equation (such as the values for the "a," "b," and "c" constants). Following that response, the student asks what the whole code should look like, so the chatbot explains that it cannot provide direct coding solutions and instead provides a walkthrough of what the code should accomplish. Later in the chatlog the student asks:

"what is the micropython code for natural log?"

The chatbot explains how to import the “math” library and to use the “log” function. The student receives an error when running the code, so they ask the chatbot what the error means, and the chatbot explains it happens with improper use of the “^” operator. After, the student asks how to write x^3 in code, and the chatbot explains that the “^” symbol must be replaced with “**” in order to indicate “to the power of” in the code. In this session, the chatbot breaks down the initial question to help work through the problem with the student and avoids giving up the answer to the student. Instead of just giving the code when the student was having trouble, the chatbot asks for more specific details to help work through the problem. The student and chatbot engage in this back-and-forth exchange to solve the smaller problems together and work their way to a functioning code.
Analysis: In this session, the student asks the chatbot for help creating and debugging code. The chatbot assists the student by providing variables used in the code as a starting place and offering to help the student through the rest of it. Further, the chatbot helps debug a syntax error in the code.

The next session involves Lab 5 where students use a strain gauge to measure the pressure of a soda can. At the end of the lab, students perform an uncertainty analysis.

Session #3
Initial Prompt: “trying to find the uncertainty analysis for the strain lab”
Discussion Description: This session starts with the student asking the chatbot for the uncertainty analysis equation. The chatbot responded with a few areas where uncertainty could be calculated, including the voltage measurements, resistance change, strain change, and pressure calculations. The student then asked for the equation for the uncertainty analysis itself. The chatbot provided the uncertainty analysis equation and explained how to apply it with the previously mentioned equations. The student asked for clarification regarding the instructions for the lab report. The instructions were to include two error sources in the uncertainty calculations. The chatbot explained that this means to consider at least two sources of error from the measurements taken in the lab. The student asked the chatbot if gain and resistance are acceptable as error factors, and the chatbot responded that they are acceptable.
Analysis: In this session, the student used the chatbot for clarifying questions regarding the uncertainty equation and its applications. The chatbot responded to the student with sources of error and the equations needed to calculate uncertainty.

The chatbot sessions between students and the chatbot varied in length, but most of the sessions observed more than 10 responses from the student user.

Quantity of Student Responses per Session	Number of Instances in First Term	Number of Instances in Second Term	Total between Both Terms
1 Response per Session	5	8	13

2-9 Responses per Session	10	28	38
10+ Responses per Session	12	17	29

Table 2: Number of student responses per session in each academic term

Discussion

Survey Data Analysis

Overall, the reception of the chatbot's implementation seemed positive, and according to Figure 3, most survey respondents thought the chatbot helped with various tasks in the class. From Table 1, the chatbot performed well in providing information to students on a wide range of question types. The surveys obtained student opinions regarding the chatbot's usage in the classroom, and based on the results, a majority of 60% said they would use this chatbot in other classes if able to. This shows that the student experience with the chatbot was beneficial and could be implemented in other classrooms as an educational tool. In terms of chatbot complexity, 62% of the surveyed students found the chatbot easy or very easy to use while no students found it hard or very hard to use. This highlights the simplistic design nature of the chatbot as the interface and interactions were easy to follow for users. Therefore, the issue of chatbot complexity can be removed as a potential limitation of student issues with the chatbot.

Chatlog Data Analysis

Based on the significant amount of class-related inquiries and situations, the chatbot was successful in providing a variety of information resembling that of a TA. This means that the chatbot is able to approach and solve many different class assignments rather than specializing in one assignment or project component. The chatbot provided troubleshooting, offered experimental design ideas, answered conceptual questions, analyzed student data, and much more. All of these student inquiries are examples of the experimental process occurring in the student's work. Because the problem solving of students can be observed, it can be seen as evidence that critical thinking is occurring. Therefore, the chatbot can be used to help students perform better in developing problem solving and critical thinking skills.

Another observation from the chatlog data is that most chatbot interactions included more than one question from the student user. This means the student had multiple follow-up questions and responses to the chatbot whether it was meant to clarify a previous point, expand on a troubleshooting situation, or ask a different assignment-related question. As seen on Table 2, both academic terms had a combined total of 38 sessions where the responses were between two and nine. Also, 29 of the sessions are instances with more than 10 responses per session meaning a majority of sessions show students who are fully engaged with the chatbot rather than asking a question and leaving the chabot. This high engagement further shows how students find the chatbot as being helpful and useful to their learning.

Future Works

Student Requests

The main two requests for improvements were better equation formatting and the ability to generate code for lab experiments. Sometimes, the chatbot responds with poor equation formatting, making it difficult to understand what the equation is supposed to be. For example, when asked about equations, the chatbot gives out equations in LateX format rather than symbolic form. The equations would be much easier to read for users if represented in symbolic form rather than LateX form.

The chatbot was specially designed to prevent code from being generated in responses, however, it was instructed to provide pseudocode and coding advice to allow students to troubleshoot their code.

Testing and Data

As mentioned previously, this study was conducted over two academic terms with around 120 total students. Not all students participated in the study and used the chatbot, but from the students that did, even less filled out the survey and expressed their opinions. The lack of data encourages future work to be done on the use of a chatbot-based educational tool. Also, understanding the effect of the chatbot on student learning needs more conclusive data.

Conclusions

This study investigated the implementation of an AI chatbot as a learning tool in an engineering classroom, focusing on its impact on student learning, problem-solving, and TA workload. The findings revealed that the chatbot aided the learning experience for many students, particularly in areas related to lab experiments and coding. Given the significant number of chatlog sessions and most surveyed participants reporting that its responses were accurate and helpful, the chatbot demonstrated its potential as a valuable educational resource. Additionally, it contributed to fostering independent problem-solving and critical thinking skills by encouraging students to engage in iterative dialogues rather than providing direct answers, as shown by the chatlogs.

Despite these successes, the study highlighted areas for improvement. Students encountered challenges with equation formatting, lack of code generation, and occasional misunderstandings of their queries. While many of these limitations were mitigated through iterative updates and back-and-forth interactions between students and the chatbot, further development is needed to make this chatbot easier for students to utilize. For instance, integrating the chatbot with lab instruments and software outputs could enable it to provide more precise troubleshooting support, mimicking the hands-on assistance of a TA who can observe the data from instruments used in the laboratory. Broader participation in future studies with this chatbot or other similar generative AI tools will allow for more robust conclusions and quicker evaluation of subsequent improvements to the learning environment.

Overall, the chatbot demonstrated the use of AI tools in an educational environment and highlighted the benefits that students can obtain by using the chatbot. The positive feedback through surveys and wide usage of the chatbot means that this tool has potential for being a substitute tool for engineering education.

References

[1] L. D. Feisel and A. J. Rosa, "The role of the laboratory in undergraduate engineering education," *Journal of Engineering Education*, vol. 94, no. 1, pp. 121–130, Jan. 2005. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2168-9830.2005.tb00833.x

[2] E. Al-Hossami, R. Bunescu, J. Smith, and R. Teehan, Can Language Models Employ the Socratic Method? Experiments with Code Debugging. 2023. [Online]. Available: https://arxiv.org/abs/2310.03210

[3] N. H. Mthombeni, R. Maladzhi, K. Moloi, T. Mashifana, M. Tsoeu, and F. Nemavhola, "Ai chatbots as inherent tools for comprehensive learning of engineering design: A systematic review," in 2023 World Engineering Education Forum-Global Engineering Deans Council (WEEF-GEDC), 2023, pp. 1–6.

[4] F. A. Bravo and J. M. Cruz-Bohorquez, "Engineering Education in the Age of AI: Analysis of the Impact of Chatbots on Learning in Engineering," Education Sciences, vol. 14, no. 5, p. 484, 2024.

[5] Y. Yang, W. Sun, D. Sun, and S. Z. Salas-Pilco, "Navigating the AI-Enhanced STEM education landscape: a decade of insights, trends, and opportunities," Research in Science & Technological Education, pp. 1–25, 2024.

[6] E. Seymour, Partners in innovation: Teaching assistants in college science courses. Rowman & Littlefield, 2005.

[7] F. Ouyang, W. Xu, and M. Cukurova, "An artificial intelligence-driven learning analytics method to examine the collaborative problem-solving process from the complex adaptive systems perspective," International Journal of Computer-Supported Collaborative Learning, vol. 18, no. 1, pp. 39–66, 2023.

[8] T. Balart and K. J. Shryock, "A Framework for Integrating AI into Engineering Education, Empowering Human-Centered Approach for Industry 5.0," in 2024 IEEE Global Engineering Education Conference (EDUCON), 2024, pp. 1–10.

[9] D. M. Buede and W. D. Miller, The engineering design of systems: models and methods. John Wiley & Sons, 2024.

[10] B. Eager and R. Brunton, "Prompting higher education towards AI-augmented teaching and learning practice," Journal of University Teaching and Learning Practice, vol. 20, no. 5, 2023.

[11] B. Shneiderman, "Human-centered artificial intelligence: Reliable, safe & trustworthy," International Journal of Human-Computer Interaction, vol. 36, no. 6, pp. 495–504, 2020.

[12] L. A. Curry, I. M. Nembhard, and E. H. Bradley, “Qualitative and mixed methods provide unique contributions to outcomes research,” Circulation, vol. 119, no. 10, pp. 1442–1452, 2009.

[13] G. Walton and M. Hepworth, “A longitudinal study of changes in learners’ cognitive states during and following an information literacy teaching intervention,” Journal of Documentation, vol. 67, no. 3, pp. 449–479, 2011.

[14] E. Sato, V. Shyyan, S. Chauhan, and L. Christensen, “Putting AI in Fair: A Framework for Equity in AI-driven Learner Models and Inclusive Assessments,” Journal of Measurement and Evaluation in Education and Psychology, vol. 15, no. Special Issue, pp. 263–281, 2024.

[15] P. Daly, M. V. Anand, C. Abel, and A. C. Sabuncu, "Industry perspectives on mechanical engineering troubleshooting," in *Proceedings of the American Society for Engineering Education Annual Conference and Exposition*, Long Beach, CA, USA, Jun. 2024.

[16] M. Golanbari and R. Garlikov, "Employing Socratic pedagogy to improve engineering students’ critical reasoning skills: Teaching by asking instead of by telling," in *Proceedings of the American Society for Engineering Education Annual Conference and Exposition*, Pittsburgh, PA, USA, Jun. 2008.