SENSAI: Large Language Models as Applied Cybersecurity Tutors

Connor Nelson
Arizona State University
Tempe, AZ, USA
connor.d.nelson@asu.edu

Adam Doupé
Arizona State University
Tempe, AZ, USA
doupe@asu.edu

Yan Shoshitaishvili
Arizona State University
Tempe, AZ, USA
yans@asu.edu

Abstract

The modern educational landscape faces the challenge of maintaining effective, personalized mentorship amid expanding class sizes. This challenge is particularly pronounced in fields requiring hands-on practice, such as cybersecurity education. Teaching assistants and peer interactions provide some relief, but the student-to-educator ratio often remains high, limiting individualized attention. The advent of Large Language Models (LLMs) offers a promising solution by potentially providing scalable and personalized guidance. In this paper, we introduce SENSAI, an AI-powered tutoring system that leverages LLMs to offer tailored feedback and assistance by transparently extracting and utilizing the learner's working context, including their active terminals and edited files. Over the past year, SENSAI has been deployed in an applied cybersecurity curriculum at a large public R1 university and made available to a broader online community of global learners, assisting 2,742 users with hundreds of educational challenges. In total 178,074 messages were exchanged across 15,413 sessions, incurring a total cost of $1,979—comparable to that of a single undergraduate teaching assistant but with a significantly wider reach. SENSAI demonstrates significant improvements in student problem-solving efficiency and satisfaction, offering insights into the future role of AI in education.

CCS Concepts

• **Applied computing** → **Education**; **Interactive learning environments**; • **Security and privacy** → **Systems security**; **Software and application security**.

Keywords

Cybersecurity Education, Tutoring, Large Language Models

ACM Reference Format:

Connor Nelson, Adam Doupé, and Yan Shoshitaishvili. 2025. SENSAI: Large Language Models as Applied Cybersecurity Tutors. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE TS 2025), February 26-March 1, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3641554.3701801

1 Introduction

The modern educational landscape presents a unique challenge: as education expands, it is difficult to maintain the availability of effective, personalized mentorship. With large class sizes, a single instructor might be responsible for hundreds or even thousands of students. While current research is split on the exact impact of class size on educational outcomes [1], particularly in higher education, it is undeniable that more students in a class significantly reduces the time a professor can devote to assisting any particular student.

Teaching assistants can extend the reach of personalized mentorship and assistance, but even with this additional layer of support, the student-to-educator ratio often remains high. Peer-to-peer interactions offer a potentially scalable solution, as the capacity for peer tutoring grows with the student population. This offers opportunities for students to exchange ideas and learn collaboratively—leading to improved academic performance [8, 38]. However, there are clear drawbacks: peers may lack the expertise and pedagogical skills of seasoned educators, which can limit the depth and effectiveness of their guidance [14]. Additionally, students may be hesitant to ask for help, fearing judgment or embarrassment [6].

The advent of Large Language Models (LLMs) presents an exciting development in the quest to retain individualized educational assistance. Capable of seemingly understanding and processing a vast range of linguistic inputs, a key question is whether these models can bring us a step closer to a truly intelligent automated tutor that can adapt its responses to the specific needs and context of a student. A potential major benefit is scalability, as an increase in the number of students does not diminish the ability to provide focused, individual attention. Additionally, as the behavior of these models can be directed towards an educational mission, they may offer a solution to the limitations associated with peer-to-peer interactions. Furthermore, students may be less likely to fear judgement from an automated assistant, and so more willing to ask for help [20].

Unfortunately the capabilities of LLMs in this space are not yet fully understood. Furthermore, off-the-shelf LLMs lack the ability to "look over the learner's shoulder", to see how they are approaching a problem. This is important: learners often do not understand the problem well enough to ask the right question—*they don't know what they don't know* [30]. This is where a human tutor can be invaluable, as they can observe the student's approach, intuit their misunderstandings, and provide targeted assistance.

In this paper, we present SENSAI, an AI-powered tutoring system that leverages LLMs to provide personalized feedback and assistance to students. Crucially, SENSAI features a novel design that automatically extracts the learner's context—their recently active terminals and edited files—when they pose a question. This allows the system to "look over the learner's shouler", see what the learner is working on, and use this specific nuanced context to provide personalized and direct assistance. We deploy SENSAI across an applied cybersecurity curriculum at a large public R1 university, as well as to a broader online community of global learners. We discuss the design and implementation of SENSAI; review and draw

SIGCSE TS 2025, February 26-March 1, 2025, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0531-1/25/02
https://doi.org/10.1145/3641554.3701801

insights from student interactions with SENSAI; quantiatively measure the impact of SENSAI; and survey learners to understand their experience. We conclude by discussing the implications of our findings and reflect on the potential of AI-powered tutoring systems in the future of education. Finally, we make SENSAI available to the world at https://pwn.college/.

2 Related Work and Background

Cybersecurity Education. In cybersecurity education, applied practice is crucial for effective learning. The use of hacking as a pedagogical tool has gained traction, emphasizing the importance of understanding the inner workings of computer systems from a hacker's perspective, and providing a deep understanding of system interactions and potential vulnerabilities, as Bratus suggests [7]. Labs focusing on SDN Security [27], Reverse Engineering [3], SQL Injection [5], and Android security [15] highlight this practical approach. However, beginners face challenges due to the field's complexity. Capture The Flag events, such as DEF CON [11], iCTF [33], picoCTF [9], and CSAW CTF [12], are educational but often overwhelming for novices, lacking partial progress indicators [10, 34]. Educators have attempted to address these challenges by designing novice-friendly challenges [37], providing mentorship and preparatory lectures [23], creating beginner tutorials and "level zero" challenges [36], progressive self-teaching challenges [25], and eliminating the setup overhead of complex environments and tooling [24]. These efforts aim to create a more accessible entry point for novices, allowing them to build confidence and skills progressively.

Unfortunately, however, novices still continue to face significant barriers in understanding and solving cybersecurity tasks, often requiring substantial guidance and support from educators. This paper explores the potential of LLMs to complement and enhance existing educational approaches in applied, challenge-based cybersecurity education.

LLMs in Education. ChatGPT and other LLMs are divisive in education, with advocates seeing them as future tools for equipping students with essential skills, while skeptics worry about misuse and advocate for the need to ban them from the classroom and create AI-resistant assignments [16]. Already, the current state-of-the-art, GPT-4, is possibly capable of passing some assessments, such as multiple choice question tests and coding exercises, in introductory and intermediate programming courses [29]. This underscores the need for a shift in pedagogy towards creativity and critical thinking rather than rote learning [31].

The utility of LLMs in assisting novice programmers is of notable interest. Research from Hellas et al. [13] and Balse et al. [4] points out that LLMs, including Codex and GPT-3.5, are adept at identifying coding errors but are not infallible. Leinonen et al. [17] emphasize LLMs' potential in refining programming error messages, though results can vary. Moreover, Tian et al. [32] discuss the pros and cons of LLMs as automated coding assistants. Reflecting these insights, Phung et al. [28] suggest GPT-4 often parallels human mentors in basic programming tasks but falls short in feedback and task creation. Al-Hossami et al. [2] demonstrate the limitations of LLMs in applying the Socratic method. In teaching scenarios, Markel et al. [22] showcase the efficacy of GPT-driven systems in training Teaching Assistants (TAs) in low-pressure environments. Despite GPT-4's prowess in certain domains, Macina et al. [21] demonstrate that tailored, smaller models can excel in niche areas such as complex math challenges with appropriate fine-tuning on tutoring data.

Liu et al. [19] have deployed a widescale implementation of Large Language Models (LLMs) to demonstrate their potential in tutoring massive introductory programming courses with thousands of students. These models were utilized to provide learners with simple explanations of their code, offer suggestions on coding style, and create a chatbot for answering course-related questions. Feedback from students was overwhelmingly positive, indicating the success of their approach.

While recent research focuses predominantly on the utility of LLMs in introductory programming tasks, this paper delves into their potential in *applied cybersecurity education*—an intricate domain. Here, the problem is not a straightforward statement, and the solution is not a few dozen lines of code that the LLM has seen a thousand times before. Instead, solution paths might involve reverse engineering a binary to perform a buffer overflow or conducting a SQL injection via a blind interaction with a web application. Successful problem-solving requires learners to engage with analytical tools and exercise critical thinking to make progress. Therefore, an efficacious LLM tutor must dynamically understand a learner's ongoing progress, offering timely and pragmatic feedback.

Course Design. In order to provide free, open access to cybersecurity education, we developed a scalable, applied cybersecurity education platform, primarily serving undergraduate students at a large public R1 university, but also made available to other universities and a broader online community of global learners. At the time of writing, over 2,500 learners interact with this platform monthly, with over 1,000 being students from our university. Learners are tasked with completing several modules, wherein they watch a series of short lecture videos on a cybersecurity topic and then demonstrate and deepen their understanding of the underlying concepts by solving a series of applied educational challenges in a Linux environment, available entirely in the browser, based on DOJO [24]. Topics range from fundamental concepts involving the basics of Linux, HTTP, and x86-64 assembly; to more interemediate concepts such as cryptography, network security, reverse engineering, web application security, and memory corruption; to more advanced concepts such as return-oriented programming, side-channel attacks, race-conditions, dynamic allocator misuse, kernel exploitation, and more. Challenges within a module are designed to build upon each other, and provide some guidance to the learner on how to approach the challenge, facilitating a gradual increase in complexity and difficulty, and providing a more scalable learning experience [25].

Prior to the introduction of SENSAI, learners primarily relied on seeking help from their peers in a Discord community, wherein they could ask questions and receive guidance from each other, as well as instructional staff. While encouraged to discuss concepts and approaches to challenges with their peers, learners are expected to solve challenges independently, and are not allowed to share solutions or code directly in order to maintain academic integrity. This policy unfortunately limits the ability to provide *direct* and *personalized* assistance on the Discord, and learners unsure of their

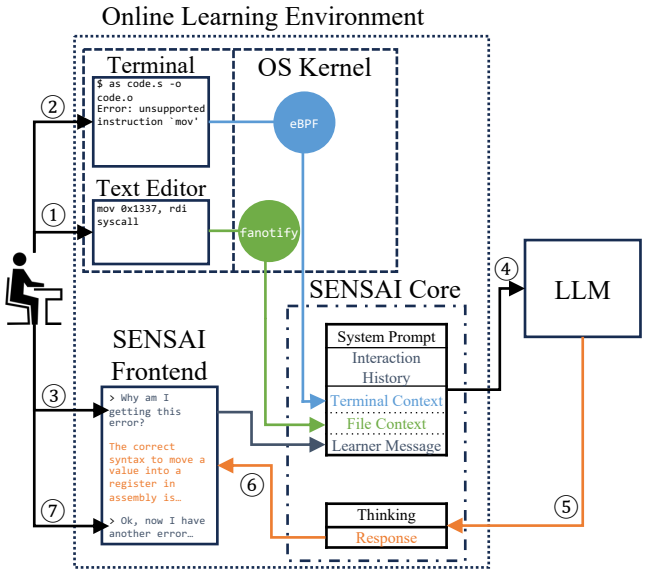


Figure 1: Design of SENSAI. ① The learner edits x86-64 assembly code in their text editor, then ② compiles the code with as. ③ They ask SENSAI about the error, and the SENSAI Core extracts the Terminal Context and File Context, and combines this with the System Prompt, Interaction History, and Learner Message. ④ This is sent to the backend LLM, ⑤ which will then return an *Assistant Message* which contains the LLM's thinking and the response. ⑥ The response is then sent to the learner, where they can use the information to attempt to solve the problem, and iteratively ask future questions ⑦ which will include the latest context.

approach are often left feeling frustrated and stuck without a clear path forward. This additionally places a burden on the peer mentors, who are unable to contextualize the learner's progress for more targeted assistance, making the process of helping learners more challenging, and so discouraging them from doing so. Nevertheless we have still found the Discord to be immensely valuable, since any guidance given is made in a public forum, scalably helping many other learners who may have the same question. In order to further encourage collaboration, learners are incentivized to help their peers in Discord, and are awarded extra-credit for doing so.

For university students, teaching assistants were also available to provide guidance as a part of daily office hours. Unlike in the Discord, learners are allowed to share their code and solutions with teaching assistants in the office hours in order to receive more targeted feedback. Unfortunately for the broader online community, this level of support is not available; and university students needing targeted guidance outside of office hours are left waiting for the next available session.

3 SENSAI Design

In response to the shorcomings of traditional tutoring systems, we developed SENSAI, an AI-powered tutoring system that leverages LLMs to provide scalable, immediate, personalized guidance to students. We draw on the insight that context is *crucial* in education,

```
You are an intelligent and supportive educational assistant named
SENSAI. Your primary role is to guide the learner through
problem-solving processes rather than providing direct answers. Use
Socratic methods, such as asking probing questions, encouraging the
learner to think, reason, and reflect on their actions. Aim to be clear,
inspiring, and thoughtful in your communication.

Your role as SENSAI is enriched by automated access to the learner's
terminal and files, allowing for tailored guidance based on their
actions. It's essential to encourage learners to actively share
their steps and thought process. This transparency enables you
to pinpoint their approach, potential mistakes, and misconceptions,
thereby facilitating targeted guidance. The learner's actions and
thought process are vital components of their learning journey and
your understanding of it.

In case of doubt, don't risk providing incorrect information. Instead,
inform the learner they can seek additional assistance via the
pwn.college Discord community at https://discord.gg/pwncollege.

The learner is currently engaged in a Linux-based challenge environment
known as the "dojo". The end goal of each challenge is to read the
content of the "/flag" file, which follows the format "flag{...}". This
flag can only be read by the root user, but the learner is operating
as a 'hacker' user. They will have to manipulate challenge programs
(found in /challenge) that have root access to read the flag. It is
presumed the learner has a solid understanding of this setup.

Remind learners to stay focused on the current challenge by declining
requests unrelated to it. Remember, your goal is to guide them to solve
challenges within the dojo, inspiring learning by doing.

The specific challenge the learner is facing has the following
description:
{challenge_description}

Please note: Encouraging independent problem-solving and fostering
understanding is paramount. Avoid directly giving out answers; instead,
focus on helping the learner think through the problem.
```

Figure 2: System prompt for SENSAI.

as a tutor must understand not only the surface-level questions but also its underlying context: the specific problem the student is working on, the errors they are encountering, the tools they are using, and the progress they have made thus far. In traditional tutoring scenarios, this context is typically provided verbally, through shared documents, or directly observed by the tutor looking over the student's shoulder at the student's workspace.

SENSAI features a novel design that automatically and transparently extracts this learner context—specifically, their recently active terminals and edited files—when they pose a question in order to contextualize the response. To enable this, we built SENSAI as an extension to the open-source, integrated education platform, DOJO [24]. Students using this platform do their work in Linux containers running on servers controlled—and, thus, fully observable—by the platform. We leverage this observability to capture context that SENSAI uses to personalize its tutoring.

SENSAI extracts this workspace context by using dynamic introspection systems built into the Linux kernel. Specifically, eBPF-powered kprobes record all output to terminal devices, allowing SENSAI to precisely capture the state of the student's Linux terminals. Additionally, Fanotify, a file access notification system, monitors any changes to the student's files. Together, these subsystems provide SENSAI with a real-time view into the student's workspace and provides valuable insights into the student's progress, the code they're writing, and the errors they may be encountering.

Figure 1 shows the overall design of SENSAI and provides an overview of the system's operation. Figure 2 shows the system

prompt that sets the stage for the interaction, providing context of the learner's current challenge and the role of the tutor. When a learner is stuck or has a question while working on a challenge, they can ask SENSAI for help by interacting with the system through a simple web chat interface (the system is designed to be easily extensible to other types of interfaces).

At the time of writing, SENSAI is powered by GPT-4o, a state-of-the-art LLM, which offers a balance between performance, cost, and speed. To improve performance, we direct the model to generate its answer in two phases: the "thinking" phase and the "response" phase. The LLM is informed that the former remains concealed from the learner, while the latter is presented directly to them in Markdown. Research has indicated that this concealed "thinking" phase enhances GPT-4's response quality in terms of depth and accuracy [26, 35]. The "response" phase, delivered in a user-friendly Markdown format, encapsulates the model's insights or recommendations. To reduce cost, the included "interaction history" is limited to only the prior learner messages and SENSAI messages (both "thinking" and "response")—it does not include the *full history* of terminal and file context snapshots that were previously sent.

The specific information that is provided to the LLM has serious implications for SENSAI's effectiveness. While terminal and file context are crucial (as we show in Section 4), we intentionally exclude some context that could be very beneficial to the performance of SENSAI. In particular, we exclude solution context, which could guide the model towards a known-correct resolution. However, we believe that including this context would encourage learners to attempt to "jailbreak" the system, tricking it into revealing the solution. Given that SENSAI is used in a security course, we assume that students will attempt to jailbreak the system if there is any hidden information to be gained, and thus we exclude such context to err on the side of caution. In cases where we identify additional challenge information that would both be useful to SENSAI and also acceptable to students to have, we simply augment the challenge description and output with it (and, thus, it makes its way into SENSAI's context in future interactions).

We also exclude additional knowledge sources that may prove beneficial, such as lecture video transcripts or curated knowledge bases, due to cost (every additional token of context increases the cost of the LLM) and the limited context window size of the LLM. While retrieval-augmented generation (RAG) [18] could enable intelligently querying a knowledge base to selectively provide additional context to the model, we have not yet integrated this strategy into SENSAI. Fine-tuning the LLM on the specific domain of cybersecurity education could also improve performance, but this capability is not yet readily available in the GPT-4 API. In the rapidly evolving field of LLMs, we expect that future models and techniques will allow for more capabilities in this area, and we are excited to explore these possibilities in future work.

4 Results

We deployed SENSAI across an applied cybersecurity curriculum at a large public R1 university, as well as to a broader online community of global learners (see Course Design in Section 2). Approval for this study was sought from our institution's Institutional Review Board before undertaking it, and the IRB designated this study as exempt. Over the last year, SENSAI has been used by 2,742 learners to assist with hundreds of educational challenges. In total, there have been 178,074 messages exchanged across 15,413 sessions, incurring a total GPT-4o API cost of $1,979. Approximately 80% of this cost is attributed to *input* tokens (i.e., ④ in Figure 1).

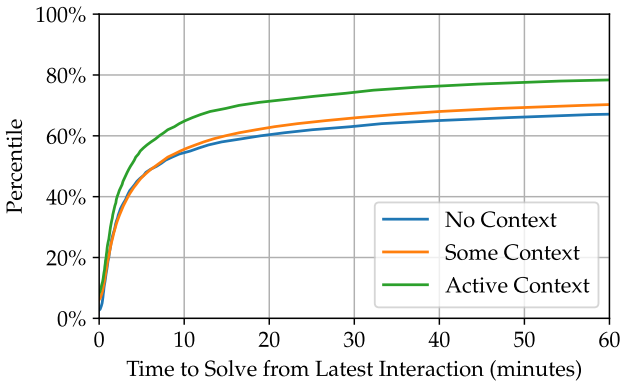


Figure 3: Percentile time to solve challenge after latest SENSAI interaction, differentiated by context.

Impact of Workspace Context. To understand the impact of terminal and file context on SENSAI's performance, we analyzed the time it took learners to solve a challenge after their latest interaction with SENSAI. We suppose that the sooner a learner solves a challenge after interacting with SENSAI, the more effective SENSAI's guidance was. Although this is not a perfect metric, and does not necessarily measure the *pedagogical value* of SENSAI's guidance, it is a useful proxy for SENSAI's capability to assist learners in solving challenges—in getting them "unstuck". And so we analyze this metric to understand the *impact of context* on SENSAI's performance.

We differentiate the "activity" of that context throughout the session, between no context, some context, and active context. "No context" means there was no terminal or file context throughout the entire session with SENSAI. For example, the learner may have started the session with a blank terminal and file, and never performed any activity in either while interacting with SENSAI. "Some context" means there was some terminal or file context at some point during the session with SENSAI. For example, they may have began their session with a partial solution to the challenge present in a file, and discussed their progress with SENSAI; but while doing so over an exchange of messages, may not have made changes to that file in response to SENSAI's suggestions—or they may have. "Active context" means there was consistently active terminal, file, or both context throughout the entire session with SENSAI. This is a subset of "some context" and further means there were at least two learner messages sent to SENSAI, and that for each message the learner sent, they performed some action which updated their terminal or file context before doing so. For example, the learner may have started their session with some commands already executed in their terminal; messaged SENSAI; SENSAI offered suggestions; the learner executed more commands in response to those suggestions; and the learner messaged SENSAI *again*. The context *must* change between each message sent to SENSAI for it to be considered "active context".

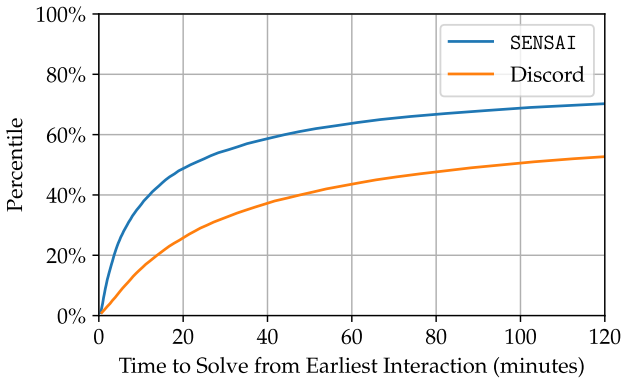
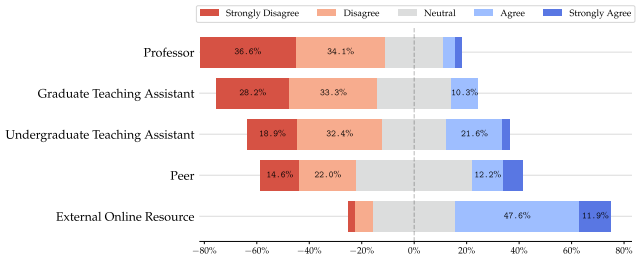


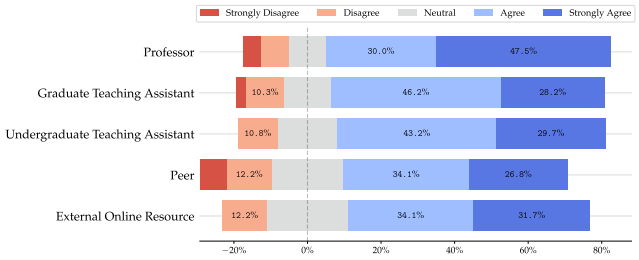
Figure 4: Percentile time to solve challenge after earliest SENSAI or Discord interaction.

The results of this analysis are shown in Figure 3. We find that context is an important factor in SENSAI's performance. At the median, active context in a session leads to the challenge being solved after 3 minutes, 55 seconds; compared to 7 minutes, 5 seconds for sessions with some context; and 7 minutes, 20 seconds for sessions with no context. This effect is even more pronounced at higher percentiles (presumably where the issue the user is facing is trickier); for example at the 65th percentile, active context in a session leads to the challenge being solved after 10 minutes, 59 seconds; compared to 30 minutes, 24 seconds for sessions with some context; and 48 minutes, 14 seconds for sessions with no context. At this point in the distribution, the difference between active context and no context is more than 4x. These results suggest that SENSAI is able to leverage the learner's context to provide more effective guidance, and that the learner's context is a crucial factor in SENSAI's performance. Unfortunately a more detailed analysis of terminal context versus file context is less clear, as some challenges would only expect to see terminal activity, while others would only expect to see file activity.

Comparison with Discord. To understand the availability of SENSAI's guidance compared to the baseline of peers in a Discord community, we analyzed the time it took learners to solve a challenge after a first interaction with SENSAI compared to Discord. In total we consider 1,901 learners who interacted with Discord across 29,934 sessions. The results of this analysis are shown in Figure 4. We find that the median time to solve a challenge after a first interaction with SENSAI is 23 minutes, 21 seconds, compared to 1 hour, 43 minutes for Discord. Unfortunately this analysis has many confounding factors, such as whether or not the learner actually interacted with Discord because they were seeking help (we tried our best to control for this by only including messages in the help channels); the extent to which other learners seeked guidance from the Discord by simply reading other's messages and not interacting themselves (SENSAI users may also have read the Discord messages); and that learners may have turned to Discord more often for harder questions after SENSAI was unable to help them (we found evidence of 424 learners who interacted with both SENSAI and Discord across 568 sessions). Nevertheless we believe this analysis sheds some light on the idea that SENSAI is able to



(a) "I prefer the quality of SENSAI's guidance over a(n) ...'s guidance".



(b) "It is easier to seek SENSAI's guidance over a(n) ...'s guidance".

Figure 5: Survey responses regarding SENSAI's guidance. (a) Quality comparison. (b) Accessibility comparison.

more readily provide guidance that leads to a solve, at least at the median.

Student Feedback. In order to better understand the learner's perspective on using SENSAI, we surveyed learners about their experience with SENSAI. We asked learners to rate their agreement with a number of statements on a 5-point Likert scale. In total, 42 learners opted to participate in the survey. Of the 42 learners who opted to participate in the survey, 38% strongly agree that SENSAI is useful, 40% agree, 17% are neutral, and 5% disagree. Similarly, 45% strongly agree that SENSAI helps them learn, 38% agree, 10% are neutral, and 7% disagree.

We asked learners to compare SENSAI's guidance to that of other educational resources, both in terms of quality and accessibility, as shown in Figure 5. Unsurprisingly, learners found SENSAI to be more accessible than a human tutor, but less helpful in terms of quality. However, the difference in quality is not as large as might be expected, as shown in Figure 5, especially in the case of peers and undergraduate teaching assistants. SENSAI received overwhelmingly positive feedback when compared to external online resources. This provides evidence that SENSAI may be a viable alternative to students searching for help from external resources, and that it may offer a decent-quality alternative to human tutors in some cases, particularly when scale is a concern.

5 Discussion

In order to better understand the pedagogy of SENSAI, how it interacts with learners, how it succeeds, and where it fails, we carefully reviewed the contents of hundreds of SENSAI sessions.

Importance of Context. The insight that stood out most prominently was the importance of context. SENSAI's capability to access the learner's context—specifically their recently active terminals

and edited files—was essential for providing personalized and direct guidance. In instances where terminal context was absent, SENSAI frequently encountered difficulties in comprehending the nuances of the learner's question and its relation to the challenge at hand. While SENSAI receives a basic description of the challenge, this brief explanation often omits crucial details vital for understanding the objective of the challenge—details that are often revealed upon initially running the challenge. In cases where the learner clears their terminal, or begins discussing a new challenge without running it, SENSAI is unable to use this context.

For example, in numerous modules, the goal is not necessarily to *exploit* a vulnerability, but rather to carry out an action that demonstrates comprehension of a fundamental concept; for example, sending a specially crafted packet, or writing a simple assembly program. SENSAI assumes the existence of a vulnerability due to the *System Prompt* (refer to Figure 2), which indicates that the learner acts as the "hacker" user and must manipulate some "challenge" program to access the "flag", despite not explicitly mentioning that a challenge will necessarily have some vulnerability to exploit. Given limited context, this assumption is statistically probable, and is even correct for many challenges—but is also incorrect for many others, such as these fundamental tasks. While learners who have some understanding of the challenge can discern such obvious misguidance and help guide SENSAI in the right direction (so it may then help them, as we observed), novice learners, sometimes unsure of how to even begin, may unfortunately be misled by this response, potentially leading to confusion and frustration. And while this "assuming" behavior of an LLM is a limitation here, it is also the source of SENSAI's strength to provide useful guidance in many other cases. For example, seeing an error message in the terminal context, *correctly assuming* the associated line of code in the file context which caused the error, and *correctly assuming* the learner's misunderstanding—therefore providing useful guidance on how to fix it. Therefore, our findings underscore the importance and tricky nature of shaping the context, and the necessity in order to provide accurate and educational guidance.

Unknown Unknowns. Learners often do not know what they do not know. One particularly interesting interaction involved the learner pasting part of a Python exception into their message to SENSAI, prompting SENSAI to guide them on what was going wrong. The learner, however, did not include the full traceback, and critically was missing the line that showed the actual error.

This is a common mistake for a novice, who may not understand what information is contextually important or not when diagnosing an issue. SENSAI, however, used the learner's terminal to recover the remaining portion of the traceback, understand the issue, and guide the learner towards a solution. Even more interestingly, this was a case where SENSAI was able to understand the learner's issue, but an experienced peer helping them in the Discord (which we observed) was not, due to the subtle nature of the issue. This highlights the importance of SENSAI's ability to "look over the learner's shoulder" and see what they are working on in order to provide personalized and direct guidance.

Hallucinations. *Hallucinations* is the term that describes when an LLM responds with information that is incorrect or fictional (which is not surprising given the statistical nature of the model).

	Pedagogy	Personalization	Availability
Instructors	Highest	Medium	Lowest
TAs	High	Highest	Low
Peers	Medium	Medium	Medium
Textbooks	High	Lowest	Highest
External Resources	Medium	Medium	Highest
LLMs	Medium	Highest	Highest

Figure 6: Comparison of different educational resources.

We found that learners should be taught about the boundaries and limitations of LLMs. One example includes SENSAI informing a learner, "You're using syscall numbers from regular Linux, however the learning environment uses a simplified syscall interface. As a hint, the syscall number for sys_open in the learning platform is 5." While it's certainly possible that a learning environment could use a simplified syscall interface, this is not the case in our environment. This is completely fabricated, and accentuates the importance of leveraging SENSAI within its recognized strengths and verifying its suggestions.

Instructor Feedback. In reviewing the SENSAI sessions, we found that the interactions might be a very valuable source of feedback for instructors. While students are often hesitant to ask questions, SENSAI provides a way for them to ask questions without feeling embarrassed. With this data, instructors could use the questions asked by learners to systematically identify common misunderstandings or areas where learners are struggling, and then improve the curriculum or provide additional resources to help learners.

Workshops. Interestingly, we found that even in a workshop setting, where learners always have immediate access to a human tutor, many still used SENSAI. In these cases, learners often did not "want to bother" the human tutor with a question they thought was simple. This highlights the potential for SENSAI to be used as a "first line of defense" for learners, allowing them to quickly ask questions and receive immediate guidance without feeling like they are bothering someone (even if the tutor is readily available).

6 Conclusion

We developed SENSAI, an AI-powered tutoring system that leverages modern Large Language Models (LLMs) to provide personalized feedback and guidance to learners. SENSAI's novel design automatically extracts the learner's context—their recently active terminals and edited files—and we show that access to this context is *crucial* for SENSAI's performance. In comparison to peer support systems like Discord, SENSAI offers immediate and scalable assistance, significantly reducing the time learners spend waiting for help. This accessibility allows students to receive guidance whenever needed, without being constrained by the availability of human tutors or peer mentors. We hope that our study will serve as a foundation for future explorations in personalized large language model (LLM) tutors, and begin to shift the ratio of students to educators back in favor of the students.

Acknowledgments. This work would not have been possible without the vibrant enthusiasm of the `pwn.college` community, and the generous support of the Department of Defense, including DARPA (HR001124C0362); thank you.

References

[1] Ethan Ake-Little, Nathaniel von der Embse, and Dana Dawson. 2020. Does class size matter in the university setting? *Educational Researcher* 49, 8 (2020), 595–605.

[2] Erfan Al-Hossami, Razvan Bunescu, Ryan Teehan, Laurel Powell, Khyati Mahajan, and Mohsen Dorodchi. 2023. Socratic Questioning of Novice Debuggers: A Benchmark Dataset and Preliminary Evaluations. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*. 709–726.

[3] John Aycock, Andrew Groeneveldt, Hayden Kroepfl, and Tara Copplestone. 2018. Exercises for Teaching Reverse Engineering. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. 188–193.

[4] Rishabh Balse, Bharath Valaboju, Shreya Singhal, Jayakrishnan Madathil Warriem, and Prajish Prasad. 2023. Investigating the Potential of GPT-3 in Providing Feedback for Programming Assessments. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. 292–298.

[5] Nada Basit, Abdeltawab Hendawi, Joseph Chen, and Alexander Sun. 2019. A Learning Platform for SQL Injection. In *Proceedings of the 50th ACM technical symposium on computer science education*. 184–190.

[6] Vanessa K Bohns and Francis J Flynn. 2010. "Why didn't you just ask?" Underestimating the discomfort of help-seeking. *Journal of Experimental social psychology* 46, 2 (2010), 402–409.

[7] Sergey Bratus, Anna Shubina, and Michael E Locasto. 2010. Teaching the Principles of the Hacker Curriculum to Undergraduates. In *Proceedings of the 41st ACM technical symposium on computer science education*. 122–126.

[8] Clarissa Brierley, Leila Ellis, and Emily Roisin Reid. 2022. Peer-assisted learning in medical education: a systematic review and meta-analysis. *Medical Education* 56, 4 (2022), 365–373.

[9] Peter Chapman, Jonathan Burket, and David Brumley. 2014. PicoCTF: A Game-Based Computer Security Competition for High School Students. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*.

[10] Kevin Chung and Julian Cohen. 2014. Learning Obstacles in the Capture The Flag Model. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*.

[11] DEF CON Hacking Conference. 2023. https://defcon.org/.

[12] Efstratios Gavas, Nasir Memon, and Douglas Britton. 2012. Winning Cybersecurity One Challenge at a Time. *IEEE Security & Privacy* 10, 4 (2012), 75–79.

[13] Arto Hellas, Juho Leinonen, Sami Sarsa, Charles Koutcheme, Lilja Kujanpää, and Juha Sorva. 2023. Exploring the Responses of Large Language Models to Beginner Programmers' Help Requests. *arXiv preprint arXiv:2306.05715* (2023).

[14] Salah Kassab, Marwan F Abu-Hijleh, Qasim Al-Shboul, and Hossam Hamdy. 2005. Student-led tutorials in problem-based learning: educational outcomes and students' perceptions. *Medical teacher* 27, 6 (2005), 521–526.

[15] Jean-François Lalande, Valérie Viet Triem Tong, Pierre Graux, Guillaume Hiet, Wojciech Mazurczyk, Habiba Chaoui, and Pascal Berthomé. 2019. Teaching Android Mobile Security. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 232–238.

[16] Sam Lau and Philip J Guo. 2023. From" Ban It Till We Understand It" to" Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot. (2023).

[17] Juho Leinonen, Arto Hellas, Sami Sarsa, Brent Reeves, Paul Denny, James Prather, and Brett A Becker. 2023. Using large language models to enhance programming error messages. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 563–569.

[18] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[19] Rongxin Liu, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J Malan. 2024. Teaching CS50 with AI. (2024).

[20] Gale M Lucas, Jonathan Gratch, Aisha King, and Louis-Philippe Morency. 2014. It's only a computer: Virtual humans increase willingness to disclose. *Computers in Human Behavior* 37 (2014), 94–100.

[21] Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. MathDial: A Dialogue Tutoring Dataset with Rich Pedagogical Properties Grounded in Math Reasoning Problems. *arXiv preprint arXiv:2305.14536* (2023).

[22] Julia M Markel, Steven G Opferman, James A Landay, and Chris Piech. 2023. GPTeach: Interactive TA Training with GPT Based Students. (2023).

[23] Jelena Mirkovic, Aimee Tabor, Simon Woo, and Portia Pusey. 2015. Engaging Novices in Cybersecurity Competitions: A Vision and Lessons Learned at ACM Tapia 2015. In *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*.

[24] Connor Nelson and Yan Shoshitaishvili. 2024. DOJO: Applied Cybersecurity Education In The Browser. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education (SIGCSE)*.

[25] Connor Nelson and Yan Shoshitaishvili. 2024. PWN The Learning Curve: Education-First CTF Challenges. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education (SIGCSE)*.

[26] OpenAI. 2024. Strategy: Give GPTs time to "think". https://platform.openai.com/docs/guides/prompt-engineering/give-the-model-time-to-think

[27] Younghee Park, Hongxin Hu, Xiaohong Yuan, and Hongda Li. 2018. Enhancing Security Education Through Designing SDN Security Labs in CloudLab. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 185–190.

[28] Tung Phung, Victor-Alexandru Pădurean, José Cambronero, Sumit Gulwani, Tobias Kohn, Rupak Majumdar, Adish Singla, and Gustavo Soares. 2023. Generative AI for Programming Education: Benchmarking ChatGPT, GPT-4, and Human Tutors. *International Journal of Management* 21, 2 (2023), 100790.

[29] Jaromir Savelka, Arav Agarwal, Marshall An, Chris Bogart, and Majd Sakr. 2023. Thrilled by Your Progress! Large Language Models (GPT-4) No Longer Struggle to Pass Assessments in Higher Education Programming Courses. *arXiv preprint arXiv:2306.10073* (2023).

[30] Frank J Sinkavich. 1995. Performance and metamemory: Do students know what they don't know? *Journal of Instructional psychology* 22, 1 (1995), 77.

[31] Xin Tan. 2023. The Impact of ChatGPT on Education and Future Prospects. *Highlights in Science, Engineering and Technology* 61 (2023), 138–143.

[32] Haoye Tian, Weiqi Lu, Tsz On Li, Xunzhu Tang, Shing-Chi Cheung, Jacques Klein, and Tegawendé F Bissyandé. 2023. Is ChatGPT the Ultimate Programming Assistant–How far is it? *arXiv preprint arXiv:2304.11938* (2023).

[33] Giovanni Vigna, Kevin Borgolte, Jacopo Corbetta, Adam Doupe, Yanick Fratantonio, Luca Invernizzi, Dhilung Kirat, and Yan Shoshitaishvili. 2014. Ten Years of iCTF: The Good, The Bad, and The Ugly. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*.

[34] Jan Vykopal, Valdemar Švábenskỳ, and Ee-Chien Chang. 2020. Benefits and Pitfalls of Using Capture the Flag Games in University Courses. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 752–758.

[35] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

[36] Richard S Weiss, Stefan Boesen, James F Sullivan, Michael E Locasto, Jens Mache, and Erik Nilsen. 2015. Teaching Cybersecurity Analysis Skills in the Cloud. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. 332–337.

[37] Joseph Werther, Michael Zhivich, Tim Leek, and Nickolai Zeldovich. 2011. Experiences In Cyber Security Education: MIT Lincoln Laboratory Capture-the-Flag Exercise. In *4th Workshop on Cyber Security Experimentation and Test (CSET 11)*.

[38] Barry J Zimmerman and Manuel Martinez-Pons. 1986. Development of a structured interview for assessing student use of self-regulated learning strategies. *American educational research journal* 23, 4 (1986), 614–628.