EDUHints: A Human-in-the-Loop Small Language Model Hint Generation System for Cybersecurity Education

Taylor Wolff[1], Richard Weiss[1], Jack Cook[1], Joseph Granville[1] and Jens Mache[2]
[1]The Evergreen State College, Olympia, Washington, USA
[2]Lewis & Clark College, Portland, Oregon, USA

taylor.wolff@evergreen.edu
weissr@evergreen.edu
cookjackc@gmail.com
jwgranville@gmail.com
jmache@lclark.edu

Abstract: The problem that we study is how to efficiently generate hints for students who are engaged in hands-on cybersecurity exercises. Students sometimes get stuck and can become frustrated when they are missing information that is necessary for solving a challenge. While large language models (LLMs) could help, they can be expensive to use and typically require the sharing of student data with third-party AI providers. In order to minimize computational overhead and financial costs, we chose to deploy a small language model (SLM) with retrieval-augmented generation (RAG). In addition, we use a human-in-the-loop approach, where the instructor reviews the AI-generated hints before they reach the student. This keeps the instructor involved, increases the quality of the hints presented to the student, and preserves student-instructor interaction while reducing the cognitive load on the instructor. We have tested our hint generation system "EDUHints" in the classroom, collecting qualitative responses from 15 students via three brief surveys.

Keywords: Cybersecurity education, Small language model, Local AI, Human-in-the-Loop

1. Introduction

EDURange is a cybersecurity education platform "designed especially for the needs of teaching faculty. The scenarios we have implemented are designed specifically to nurture the development of analysis skills in students as a complement to both theoretical security concepts and specific software tools" (Weiss et al, 2017). EDURange3's source code and build instructions are publicly available at github.com/edurange/edurange3. Students can connect and participate via the application's web user interface and terminal. The application features an integrated chat messaging system to communicate with the instructor directly. Scenarios are cybersecurity exercises with both tasks to complete and questions to answer. Each student's command line activity, chat messages, and question responses are logged and viewable by the instructor via a user interface.

Students may need assistance while completing scenarios. Thus far, the primary method instructors employed to assist students was the manual composition and sending of hints via application's chat messaging system. While instructors found that this method could be effective, it was time-consuming. The slowness of this process resulted in some students not receiving enough assistance, especially in larger classes. To address this problem, we sought to investigate and develop a new system that could improve the efficiency with which an instructor can assist students. In this paper, we present 'EDUHints', a human-in-the-loop small language model hint generation system for cybersecurity education. We developed EDUHints with four objectives in mind:

- O1. Increase the efficiency at which an instructor can provide hints to students.
- O2. Improve instructor workflow while preserving organic student-instructor interaction.
- O3. Provide a cost-effective alternative to paid third-party large language model APIs.
- O4. Preserve student data privacy.

This paper begins with a discussion on generative AI technologies, followed by an overview of the system's human-in-the-loop design, current user interface, capabilities, early testing methodologies, results, and discussion. We conclude with a review of our objectives and plans for future work.

2. Prior Work: Generative AI Technologies

Previous research efforts regarding the development and testing of AI systems that assist students have primarily focused on the integration of LLMs (Hellas, Leinonen, and Leppänen, 2024; Salminen et al, 2024; Lieb and Goel, 2024; Neumann et al, 2024). Yet, there are many drawbacks to LLMs. They can use excessive amounts of computing power, be costly for the student or school, and potentially require the sharing of sensitive student data with a third-party AI provider. Yu et al (2025) propose a possible alternative for educational software, the

small language model (SLM). SLMs are language models with less computational complexity than their larger counterparts. Lu et al (2024) note that "The vision behind SLMs is to democratize access to machine intelligence, making it both accessible and affordable to people everywhere." While SLMs typically perform worse than LLMs, the emerging technique of retrieval-augmented generation (RAG) has shown promise in enhancing an SLM's performance such that they outperform some larger models (Liu et al, 2024). Because of their reduced computational requirements, many SLMs can thus be locally deployed, running on even modest consumer CPUs (ggml-org, 2025). The SLM appears to have potential as a lower cost, more private, and reasonably performant alternative to third-party LLM APIs.

3. Human-in-the-Loop Design

With the rapid adoption of AI tools, its potential to enhance and/or replace the role of educators remains debated. Work by Chan and Tsi (2024) concluded in part that "There appears to be a consensus among teachers and students that generative AI cannot replace the human qualities of teachers that are essential for facilitating students' generic competency development and personal growth. However, as generative AI is set to become a prominent feature in many areas of everyday life, higher education institutions need to rethink how curricula can be designed to capitalize on the human qualities of teachers and the potential of generative AI to transform learning. Ultimately, creating a synergy between humans and technology is key to success in an AI-dominated world."

Our primary concern is the potential for students to develop an over-reliance or over-dependence on these tools. Zhai, Wibowo, and Li (2024) examine this topic extensively, noting that students' over-dependence on AI dialog systems may impact many of their "essential cognitive abilities, including decision-making, critical thinking, and analytical reasoning." We have concerns that such an over-dependence may lead to fewer students seeking interaction with their instructors. To sidestep the problem of over-reliance altogether, we developed a human-in-the-loop system where instructors alone are provided access to AI tools. From the system's dashboard, instructors can request, edit, and send AI-generated hints directly to students participating in EDURange scenarios. The instructor also serves as a moderator of any and all AI-generated content, which, despite preventative guardrails, can still contain factual errors, hallucinations, and offensive content.

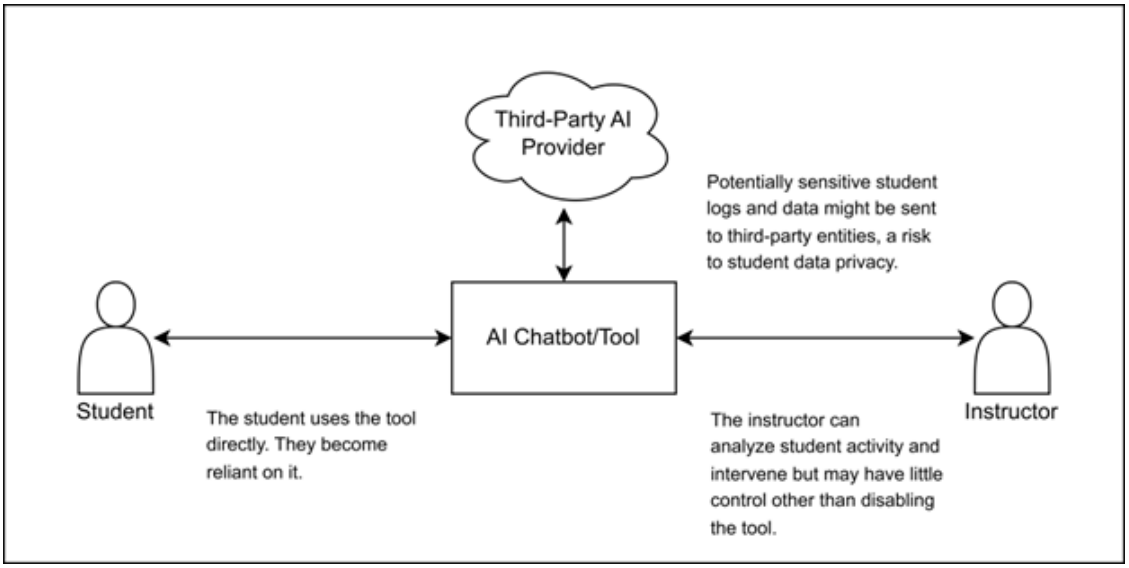

Figure 1: User diagram of a typical AI classroom assistant. Students may prioritize asking the AI tool before the instructor, possibly reducing student-instructor interaction

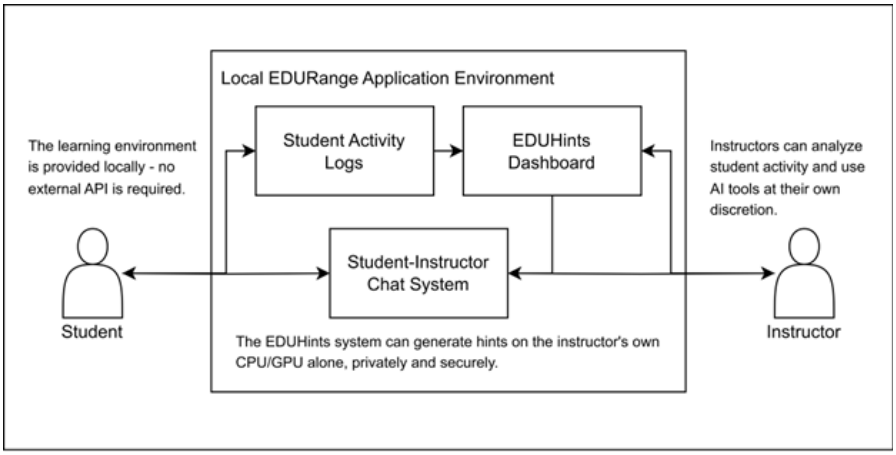

Figure 2: User diagram of the EDUHints system. Students don't use the AI tool directly, preserving student-instructor interaction

4. User Interface

For its user interface, EDUHints features a dashboard consisting of three tools: the log viewer, settings menu, and hint manager.

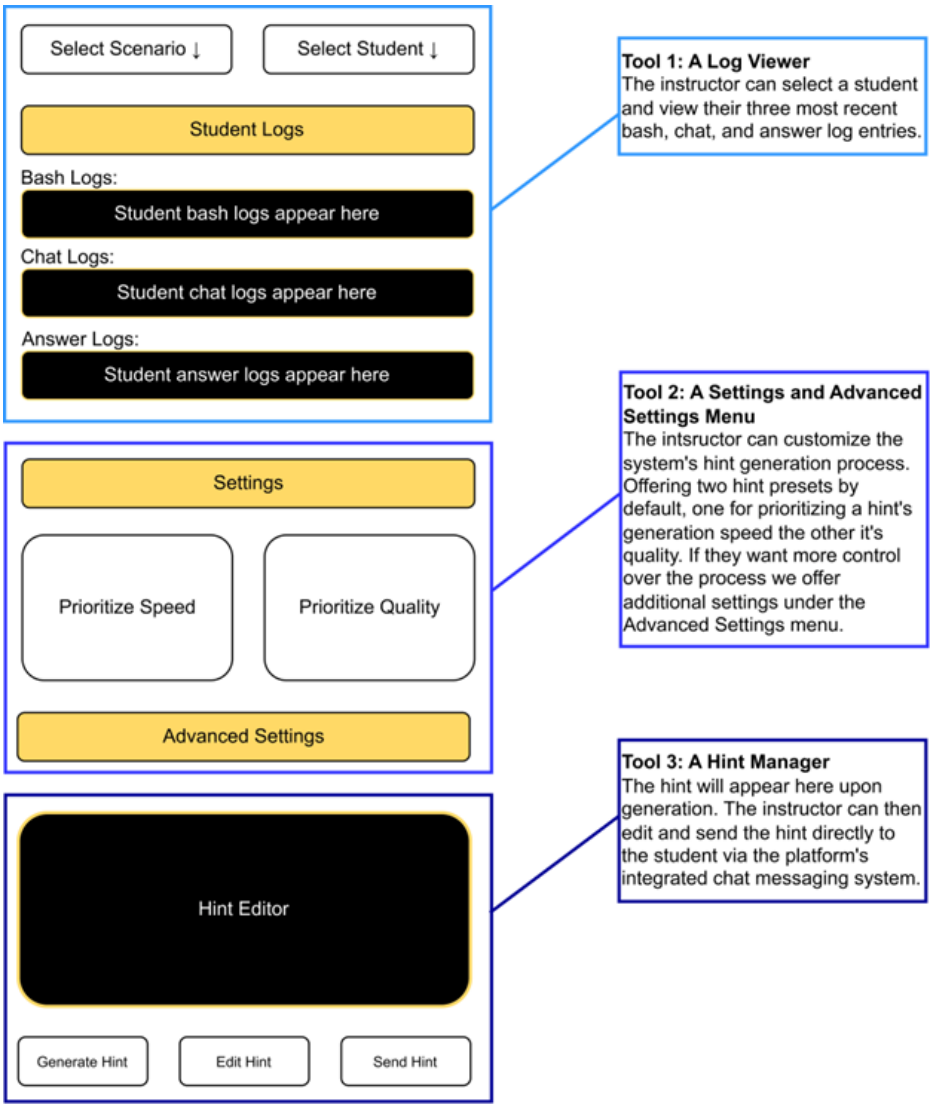

Figure 3: Diagram of the EDUHints dashboard UI

5. Capabilities

5.1 Deploy EDUHints Locally or in the Cloud

The EDUHints system leverages the reduced size and complexity of SLMs to generate hints on the CPU and/or GPU of the EDURange processes' host machine with the llama.cpp library (ggml-org, 2025) and its Python language bindings (abetlen, 2025). To do so, a machine with at least 8GB of RAM, an 8-core CPU, and 40GB of storage is recommended.

Because EDUHints is currently a subsystem of the EDURange application, it must run on the same device as its parent EDURange process, whether that process runs on the instructor's computer, a cloud-based virtual machine, or anywhere else. While the system's functionality has been tested extensively on local hardware, for classroom tests, we decided to avoid complications with our respective institution's IT infrastructure for now by running EDURange/EDUHints on a private and affordable e2-highcpu-8 Google Cloud virtual machine with eight Virtual CPUs and 8GB of RAM (Google, 2025). Once all infrastructural details are assessed, our goal is to run EDURange/EDUHints exclusively on the instructor's own local hardware for all future full live classroom tests of the EDUHints system.

5.2 Phi-3 and Model Compatibility

The default SLM for EDUHints Version 1.0 is the open-source and permissively licensed (MIT license) Phi-3-Mini-4k-Instruct language model from Microsoft. Phi-3 is a 3.8 billion parameter language model who's performance benchmarks rival those of some LLMs (Abdin et al, 2024). While Phi-3-Mini-4k-Instruct is the only model tested thus far, any model compatible with llama.cpp should also be compatible with our system. We want to verify this in future work.

5.3 Retrieval-Augmented Generation

Upon the request of a hint, EDUHints leverages retrieval-augmented generation (RAG) to enhance the hint's prompt with the selected student's recent log data and an optional "Scenario Context File." This simple text file provides a high-level summary of the particular scenario's challenges, themes, and learning objectives. We believe that hints generated with this additional file may prove more relevant to students.

5.4 Generation Times

The average generation time for a hint can vary depending on its generation settings and the target student's recent logs. For one classroom test with the "Treasure Hunt" scenario where various generation settings were selected throughout, the average generation time for a hint was 38.25 seconds.

6. Classroom Tests

In order to gauge students' overall experience with the first public test version of EDUHints, three classroom tests were conducted at two small liberal arts colleges in the United States. Two of these tests occurred within introductory computer science courses and one in an advanced computer science course. Each test was approximately an hour in duration. The EDUHints system was operated by one developer who had limited cybersecurity knowledge and instructional experience. The operator monitored the chat and log activity of students to identify which needed help, either expressly through chat messages or by indicators such as repeated failed attempts at answering an exercise question. Hints were edited for length or grammar, with off-topic hints omitted. An optional, anonymous Google Forms survey was provided at the end of each test. This survey first asked participants whether or not they received an EDUHint, and if so, to then rate their agreement or disagreement with three statements on a Likert-type scale with five choices ranging from "Strongly Disagree" to "Strongly Agree".

7. Results

Approximately 60 students participated in the classroom tests; from these, 15 students received EDUHint(s), all of which provided a response to each statement for a total of 45 statement responses.

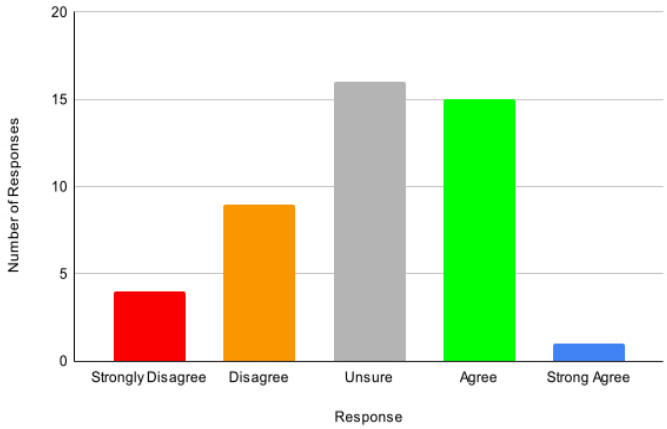

Figure 4: Bar chart for aggregate of responses to survey statements (n = 45)

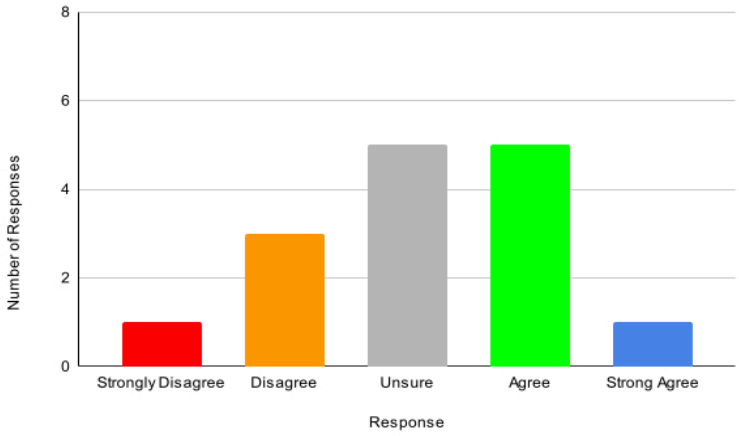

Figure 5: Bar chart representation of student responses to statement 1: "The EDUHints(s) I received were easy to understand" (n = 15)

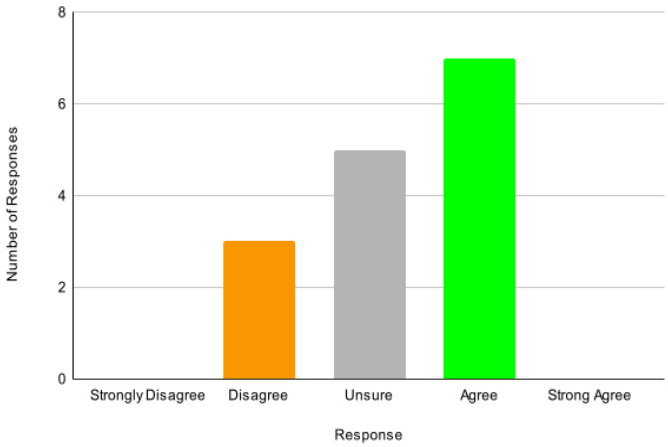

Figure 6: Bar chart representation of student responses to statement 2: "The EDUHints(s) I received were relevant to what I was working on" (n = 15)

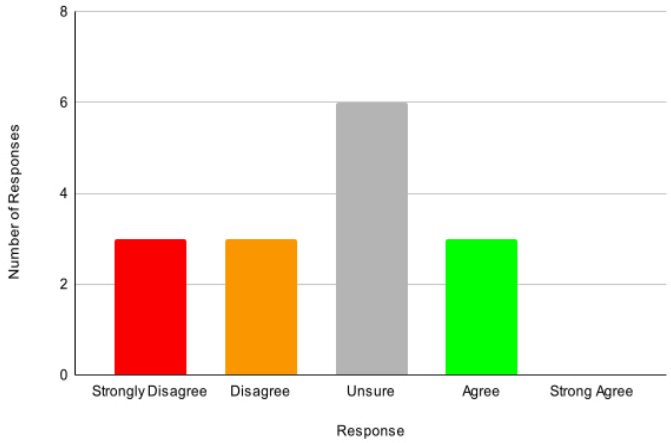

Figure 7: Bar chart representation of student responses to statement 3: "The EDUHints(s) I received helped me to solve problem(s) I was stuck on" (n = 15)

8. Discussion

While early survey results show that more respondents agree than disagree with statements 1 and 2, responses to statement 3 and the high proportion of unsure responses indicate an overall mixed or unsure response from students. We suspect that the system currently generates hints of an inconsistent quality. In order to investigate the variability of their quality, we plan to develop a quantitative process for evaluating the quality of generated hints. Such a high proportion of unsure responses may also indicate a need to reevaluate our currently subjective surveying methods.

Our work faced several limitations, most notably a small number of test participants, which we hope will increase in the near future through efforts of academic collaboration and involvement with both local and national cybersecurity education communities. Another challenge has been the technical limitations of EDURange's current data logging system, which is undergoing a significant redesign.

9. Conclusion and Future Work

We have developed and tested a human-in-the-loop small language model hint generation system for cybersecurity education. Amongst our four objectives, we have thus far accomplished objectives two, three, and four via the system's human-in-the-loop design and local deployment capabilities. For objective one, further improvements to the system are desired, such as methods for evaluating the quality of generated hints, reducing generation times, testing the performance of newer models such as Phi-4 (Abdin et al, 2024), model fine-tuning, and extending the work of Birdwell et al (2024) and Švábenský et al (2022) by developing a classifier for automatically identifying which EDURange students may need help.

Acknowledgements

This work was partially supported by the National Science Foundation under Awards 2216485 and 2216492 and the Evergreen Summer Undergraduate Research Fellows program.

Ethics Declaration: The Institutional Review Board of Lewis & Clark College has determined that this project is exempt under exemption code 45 CFR 46.104(d)(1) and will not require further review or approval from the committee.

AI Declaration: AI tools were used to generate the contents of some prototypal Scenario Context Files and to assist in the writing of some code.

References

Abdin, M., Aneja, J., Awadalla, H., Awadallah, A., Awan, A.A., Bach, N., Bahree, A., Bakhtiari, A., Bao, J., Behl, H. and Benhaim, A., 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*. https://doi.org/10.48550/arXiv.2404.14219

Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R.J., Javaheripi, M., Kauffmann, P. and Lee, J.R., 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*. https://doi.org/10.48550/arXiv.2412.08905

abetlen (2023). *Python Bindings for llama.cpp*. [online] GitHub. Available at: https://github.com/abetlen/llama-cpp-python [Accessed 21 May 2025].

Birdwell, A., Cook, J., Weiss, R., and Mache, J. (2024) "From Logs to Learning: Applying Machine Learning to Instructor Intervention in Cybersecurity Exercises" Paper presented at 2024 ASEE Annual Conference & Exposition, Portland, Oregon. 10.18260/1-2—46864

Chan, C.K.Y. and Tsi, L.H., 2024. Will generative AI replace teachers in higher education? A study of teacher and student perceptions. *Studies in Educational Evaluation*, *83*, p.101395. https://doi.org/10.1016/j.stueduc.2024.101395

ggml-org (2025). *GitHub - ggml-org/llama.cpp: LLM inference in C/C++*. [online] GitHub. Available at: https://github.com/ggml-org/llama.cpp [Accessed 21 May 2025].

Google (2025). *Cloud Build pricing*. [online] Google Cloud. Available at: https://cloud.google.com/build/pricing [Accessed 21 May 2025].

Hellas, A., Leinonen, J. and Leppänen, L., 2024, December. Experiences from integrating large language model chatbots into the classroom. In *Proceedings of the 2024 on ACM Virtual Global Computing Education Conference V. 1* (pp. 46-52).https://doi.org/10.1145/3649165.3690101

Lieb, A. and Goel, T., 2024, May. Student interaction with NewtBot: An LLM-as-tutor chatbot for secondary physics education. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (pp. 1-8).https://doi.org/10.1145/3613905.3647957

Liu, S., Yu, Z., Huang, F., Bulbulia, Y., Bergen, A. and Liut, M., 2024. Can Small Language Models With Retrieval-Augmented Generation Replace Large Language Models When Learning Computer Science?. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1* (pp. 388-393). https://doi.org/10.1145/3649217.3653554

Lu, Z., Li, X., Cai, D., Yi, R., Liu, F., Zhang, X., Lane, N.D. and Xu, M., 2024. Small language models: Survey, measurements, and insights. *arXiv preprint arXiv:2409.15790*. https://doi.org/10.48550/arxiv.2409.15790.

Neumann, A.T., Yin, Y., Sowe, S., Decker, S. and Jarke, M., 2024. An llm-driven chatbot in higher education for databases and information systems. *IEEE Transactions on Education*. https://doi.org/10.1109/TE.2024.3467912.

Salminen, J., Jung, S.G., Medina, J., Aldous, K., Azem, J., Akhtar, W. and Jansen, B.J., 2024, July. Using cipherbot: An exploratory analysis of student interaction with an llm-based educational chatbot. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale* (pp. 279-283).https://doi.org/10.1145/3657604.3664690.

Švábenský, V., Weiss, R., Cook, J., Vykopal, J., Čeleda, P., Mache, J., Chudovský, R. and Chattopadhyay, A. (2022) "Evaluating Two Approaches to Assessing Student Progress in Cybersecurity Exercises", In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education (SIGCSE 2022), March 3–5, 2022, Providence, RI, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3478431.3499414

Weiss, R., Turbak, F., Mache, J., and Locasto, M. (2017) "Cybersecurity Education and Assessment in EDURange", IEEE Security & Privacy, vol. 15, pp. 90-95, https://www.computer.org/csdl/mags/sp/2017/03/msp2017030090-abs.html

Yu, Z., Liu, S., Denny, P., Bergen, A. and Liut, M., 2025, February. Integrating Small Language Models with Retrieval-Augmented Generation in Computing Education: Key Takeaways, Setup, and Practical Insights. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (pp. 1302-1308). https://doi.org/10.1145/3641554.3701844.

Zhai, C., Wibowo, S. and Li, L.D., 2024. The effects of over-reliance on AI dialogue systems on students' cognitive abilities: a systematic review. *Smart Learning Environments*, *11*(1), p.28. https://doi.org/10.1186/s40561-024-00316-7.