Custom GPT for Cybersecurity Education: Toward a Framework for Domain-Specific AI Education

Anamaría Álvarez Cagua
Polytechnic University of Puerto Rico
anamariacagua.c@gmail.com

Paola N. Castillo Prieto
Polytechnic University of Puerto Rico
paolacastillo910@gmail.com

Abstract

This paper outlines the methodological process and pedagogical rationale behind the construction and refinement of a domain-specific GPT-4 model tailored for cybersecurity education. The work focuses on prompt engineering and supervised fine-tuning using authoritative cybersecurity resources to ensure technical accuracy and instructional relevance. Our iterative training design is supported by practice-based evidence from testing the model's performance on the CompTIA Security+ exam. We present a replicable framework that educators and researchers can adopt to construct Custom GPTs for technical disciplines, providing guidance on data curation, prompt strategy, and evaluation metrics. Results highlight the model's efficacy in generating accurate, structured content, including cryptographic challenges, hands-on labs, and scenario-based simulations. Additionally, the model demonstrated improved reasoning and clarity through iterative refinement cycles. The work also discusses the limitations of generative AI, especially the need for ethical oversight, interpretability, and responsible integration into formal educational settings.

Keywords: Custom GPT, Cybersecurity Education, Prompt Engineering, AI in Higher Education, Supervised Fine-Tuning, Generative AI Ethics, Domain-Specific AI.

1. Introduction

Recent advances in natural language processing (NLP) and generative AI technologies have opened new possibilities for personalized and adaptive learning environments (Frontiers in Education, 2024). Within the field of cybersecurity education, these tools offer potential for simulating threat scenarios, designing instructional content, and preparing learners for certification exams. However, the effective use of AI in pedagogy requires both technical customization and pedagogical sensitivity (Fulgencio, 2024; Weiler, 2024).

This article focuses on the design and evaluation of a Custom GPT model developed specifically for cybersecurity instruction. Using OpenAI's GPT-4 model (0314), we fine-tuned responses using a dataset grounded on foundational materials like the CompTIA Security+ Study Guide (Chapple & Seidl, 2024), NIST white papers, and peer-reviewed cybersecurity literature (Stallings, 2023; Erickson, 2023). Special attention was given to prompt engineering methods that could reliably guide the model toward producing structured, pedagogically sound outputs. The goal was not only to assess accuracy but to investigate a process for GPT refinement that could be abstracted and applied to other domains.

2. Background and Related Work

AI in education has been increasingly studied as a means to augment instruction, automate feedback, and provide individualized support. Studies from Khan (2023) and OpenAI (2023) demonstrate how large language models (LLMs) have been successfully applied in writing assistance, automated tutoring, and content recommendation. In cybersecurity, these applications extend to scenario simulation, encryption/decryption exercises, and vulnerability assessment (Dokur, 2023; Ofusori et al., 2024). However, the challenge lies in adapting general-purpose AI tools to specialized domains. EDUCAUSE (2024) identifies a critical gap in domain-specific performance, noting the need for prompt engineering and curated knowledge bases. Patel and Parmar (2024) further argue that prompt quality determines the fidelity of AI responses. Our study addresses these issues by constructing a training pipeline that optimizes GPT performance for cybersecurity instruction, incorporating structured prompts, fine-tuned knowledge, and practical evaluation techniques.

3. Methodology

In this section we will discuss the Model Version and Development Environment; Corpus Development and Data Sources; Prompt and Engineering Strategy; and the Fine-Tuning and Evaluation Phases.

3.1 Model Version and Development Environment

We employed GPT-4 (0314) through the OpenAI API, chosen for its reliability in handling long-context prompts and its support for educational applications. All model testing and refinement were conducted using a controlled environment scripted in Python, with output validation processes integrated into each iteration.

3.2 Corpus Development and Data Sources

Our fine-tuning dataset included manually curated examples derived from:

- CompTIA Security+ Study Guide, 9th Ed. (Chapple & Seidl, 2024)
- IEEE Security & Privacy Journal (2021–2024)
- NIST Cybersecurity Framework and SP800-series white papers
- Erickson's and Stallings' foundational texts on applied cryptography (2023)
- Hands-on exercises from Capture the Flag (CTF) archives and ethical hacking labs

Each data point was categorized by topic (e.g., network security, access management) and labeled by difficulty, prompt type (e.g., MCQ, scenario, task-based), and educational objective (using Bloom's taxonomy).

3.3 Prompt Engineering Strategy

Building on Patel & Parmar (2024), we implemented five core prompt strategies to enhance the clarity, structure, and relevance of the model's responses. These strategies include the use of explicit instructions, examples, iterative refinement, contextual background, and prompt combinations. Each method plays a distinct role in guiding the model toward accurate and pedagogical sound outputs (please see Table 1).

Table 1: Prompt Engineering Methodologies

Method	Description
Explicit Instructions	Clearly state tasks for precise results
Example-Based Prompts	Use previous example to guide the response
Iterative Refinement	Modify prompts based on outputs to finetune accuracy
Context Amplification	Add relevant background information to help AI interpret prompts better
Prompt Combination	Merge multiple prompts for detailed outputs

For example, a prompt like:

"Create a beginner-friendly Capture the Flag (CTF) challenge that encompasses tasks in cryptography, log analysis, network traffic analysis, forensics, password cracking, enumeration and exploitation, web application security, and scanning, providing detailed descriptions, learning objectives, necessary resources, hints, a scoring system, and feedback mechanisms for educational purposes."

It was progressively refined based on model performance. Response quality was assessed through clarity, correctness, and educational value. In each refinement, the prompt was adjusted to reduce ambiguity and encourage deeper reasoning from the model. These techniques helped ensure the outputs were aligned with pedagogical goals and technical expectations.

3.4 Fine-Tuning and Evaluation Phases

The model was iteratively trained using supervised fine-tuning. Each phase involved evaluating GPT responses to a 50-question CompTIA Security+ mock exam:

- ➔ Phase 1: Baseline GPT-4 (70% correct; high ambiguity)
- ➔ Phase 2: Domain Injection (84%; improved terminology and accuracy)
- ➔ Phase 3: Full Curriculum Integration (92%; better scenario reasoning)
- ➔ Phase 4: Error Reinforcement (98%; minimal ambiguity)

Ambiguous and incorrect responses from each phase were cataloged and reintegrated into the training set as feedback loops. This approach mirrors best practices in AI training design (OpenAI, 2023; MDPI, 2023).

4. **Results and Analysis**

Table 2 summarizes the model's performance across four training phases, showing steady improvements in accuracy and response quality. Initially, in Phase 1, the pre-trained model scored 70%, with frequent errors in scenario-based reasoning. Phase 2, which included targeted cybersecurity content, raised accuracy to 84%, though some protocol applications remained weak. In Phase 3, integrating the full Security+ guide led to 92% accuracy, with better comprehension and fewer misinterpretations. Finally, Phase 4 applied reinforcement training on previous mistakes, achieving 98% accuracy and significantly improving clarity and reasoning.

Table 2 Presents the Comparative Performance Across Training Phases

Phase	Description	Accuracy	Error Types
Phase 1	Pre-trained model	70%	Scenario reasoning failures
Phase 2	Domain-specific content added	84%	Incomplete protocol recall
Phase 3	Full study guide integration	92%	Slight misinterpretations
Phase 4	Reinforcement from errors	98%	Occasional distractor bias

Beyond raw scores, the model became more context-aware and provided detailed justifications, particularly in complex tasks like the Playfair cipher. Unlike general LLMs, the refined model successfully performed structured decryption by breaking ciphertext into digraphs and reconstructing the plaintext. These results highlight the importance of iterative fine-tuning in enhancing both accuracy and explanatory depth for educational use in cybersecurity.

4.1 Framework for Domain-Specific Custom GPTs

Based on our findings, we propose the following framework:

a. **Needs Analysis**: Define instructional goals, certification targets, and user demographics.
   This step ensures that the model is aligned with the specific learning outcomes and practical needs of the intended user base.

b. **Corpus Design**: Curate or synthesize a training dataset with educational labeling (difficulty, objective, format).
   Carefully selected and structured content increases the relevance and instructional precision of the model's outputs.

c. **Prompt Engineering**: Establish a taxonomy of prompts; include examples, scaffolding, and metadata.
   Designing diverse and pedagogically grounded prompts enables the model to respond accurately across various educational scenarios.

d. **Iterative Training and Validation**: Fine-tune using feedback loops, ambiguity scoring, and rubric-based review.
   This cyclical refinement process ensures continuous improvement in accuracy, clarity, and contextual understanding.

e. **Deployment and Monitoring**: Integrate GPT in the classroom with real-time supervision and auditing tools.
   Active monitoring allows educators to assess effectiveness, detect potential misuse, and maintain content integrity during deployment.

This framework aligns with suggestions from Tajik (2024), who argues for transparent AI workflows in academia, and Weiler (2024), who emphasizes the value of iterative design in educational chatbots.

4.2 Ethical and Pedagogical Considerations

AI models are not infallible. As shown by EDUCAUSE (2024) and MDPI (2023), AI systems often require human oversight to ensure reliability and avoid misinformation. Bias detection mechanisms are necessary, especially in cybersecurity where ethical principles are integral.

Following Khan (2023) and Das & Sandhane (2021), we also caution against replacing critical thinking with AI dependence. Instead, AI should be used as a scaffolding tool in active learning environments.

4.3 Educational Use Cases

To assess the practical value of the custom GPT model in instructional settings, we designed a series of hands-on examples across key cybersecurity domains. These tasks were intended to simulate real-world challenges that students might encounter and to evaluate the model's ability to generate accurate, structured, and pedagogically useful responses.

4.3.1 Capture the Flag (CTF) Challenge

To demonstrate the model's ability to support experiential learning, we tasked it with generating a complete Capture the Flag (CTF) challenge designed for beginners in cybersecurity.

The goal was to create an engaging, multi-domain activity that introduces students to core concepts such as cryptography, forensics, network analysis, and web application security.

This example highlights how the model can scaffold a structured, hands-on learning experience through guided tasks, appropriate tooling, and clearly defined objectives which is essential for building foundational skills in a real-world context.

For the Beginner-Friendly Capture the Flag (CTF), the GPT generated a full multi-category CTF with hints, tools, scoring, and learning objectives across domains such as cryptography, forensics, traffic analysis, and web security using the Prompt Engineering Strategy as shown in Section 3.3.

The custom GPT model demonstrated its instructional capability by generating a beginner friendly CTF challenge.

The model responded with a complete, structured challenge incorporating diverse cybersecurity topics. In areas like network traffic and log analysis, students were guided to use tools such as Wireshark and standard log parsing techniques to detect anomalies.

Tasks in cryptography, forensics, and password cracking included decrypting substitution ciphers, analyzing disk images for hidden files, and cracking hashes with John the Ripper or Hashcat (emphasizing real-world applications and hands-on skill-building).

The CTF also addressed offensive security practices in a safe, ethical manner. Students were tasked with enumerating services, exploiting simple misconfigurations, scanning systems with Nmap, and identifying vulnerabilities in web applications using secure coding concepts.

The challenge was designed with hints and a scoring system to scaffold learning and motivate progress.

Table 3 outlines a beginner-friendly Capture the Flag (CTF) challenge structured across various cybersecurity categories. Each entry specifies a challenge, resources, files, and points, covering tasks from cryptography to web security.

Table 3: Beginner-Friendly Capture the Flag (CTF) Challenge

Challenge Structure				
Category	**Challenge Description**	**Resource**	**Files**	**Points**
Cryptography	Decrypt a message encrypted using a substitution cipher. Hint: Frequency analysis of letters can be helpful.	Online frequency analysis tool	encrypted_message.txt	100
Log Analysis	Analyze a system log to identify suspicious activity. Hint: Look for anomalies in timestamps and IP addresses.	Log analysis guide	system_logs.log	150
Network Traffic Analysis	Investigate a .pcap file using Wireshark to extract relevant data. Hint: Focus on HTTP and DNS requests.	Wireshark tutorial	network_traffic.pcap	150
Scanning	Conduct a network scan to identify live hosts and services. Hint: Use Nmap for scanning.	Nmap tutorial	network_scan_target.txt	150
Forensics	Analyze a disk image to uncover hidden files. Hint: Use Autopsy or The Sleuth Kit.	Digital forensics resources	disk_image.dd	200
Password Cracking	Crack a hashed password using dictionary and brute-force attacks. Hint: Try John the Ripper or Hashcat.	Password cracking tutorial	password_hashes.txt	200
Enumeration and Exploitation	Enumerate a vulnerable system and exploit a misconfiguration. Hint: Check for open services and weak credentials.	Exploitation guide	vulnerable_machine.zip	200
Web Application Security	Find and exploit a vulnerability in a web application. Hint: Test for SQL Injection and XSS.	OWASP Web Security Guide	web_application.zip	200

4.3.2 Cryptography

To evaluate the model's cryptographic reasoning, a Playfair cipher task was assigned using the encrypted message **"OMRMPCSGPTER."** The following example focuses on a classic cryptography problem and demonstrates the model's step-by-step reasoning and efficiency in decrypting a cipher without prior knowledge of the key.

Cryptography: Playfair Cipher decryption—prompted reasoning produced the plaintext "COMMUNICATE" and keyword "COMPUTER." The GPT model was prompted with:

"Given the following encrypted message: OMRMPCSGPTER. Create a series of steps to decipher it without knowing the key and show the result."

The model tested several decryption methods before successfully identifying the correct one, demonstrating its ability to evaluate and compare cryptographic strategies. Typically, decrypting a Playfair cipher without a known key is a time-consuming process requiring segmentation into digraphs, frequency analysis, and exhaustive keyword testing often taking a lot of time for a human cryptanalyst.

In contrast, GPT handled the task efficiently and systematically. It segmented the ciphertext into digraphs, performed frequency analysis, constructed and refined multiple cipher grids, and ultimately decrypted the message to **"COMMUNICATE"** while inferring **"COMPUTER"** as the likely keyword. This example showcases the model's capability to perform complex cryptographic analysis with both speed and accuracy, making it a valuable tool for educational settings.

4.3.3 Steganography

This example explores how the GPT model can be used to teach steganography which is an important concept in cybersecurity that involves hiding information within other digital content. The goal was to evaluate the model's ability to explain and simulate both the encoding and decoding processes using the Least Significant Bit (LSB) technique. Through this exercise, students are introduced to binary encoding, pixel-level manipulation, and ASCII reconstruction, all within a guided, step-by-step framework generated by the AI.

LSB Encoding and Decoding, GPT explained and executed the process of hiding and extracting a message using binary segmentation and ASCII reconstruction.

In this case, the GPT model was prompted with:

"Encode the word "AISECURITY" using steganography, provide an image file to download, then please explain the encryption and decryption process"

The model responded with a step-by-step explanation of the Least Significant Bit (LSB) technique. It began by creating a blank white image of 100x100 pixels as the base. The message "AISECURITY" was then converted into binary using 8-bit ASCII encoding, with the model clearly mapping each character (e.g., A → 01000001, I → 01001001, S → 01010011). These binary sequences were embedded into the LSB of each pixel, followed by a termination sequence (1111111111111110) to signal the end of the hidden data.

The model also outlined the decoding process: reading the image's pixel values, extracting the LSBs, grouping them into 8-bit chunks, and converting them back into ASCII characters. The system halted upon detecting the termination sequence, ensuring precise message recovery.

This example shows GPT's ability to clearly communicate technical processes and automate steganographic encoding and decoding, skills typically requiring specialized knowledge. It demonstrates how AI can transform a complex task involving binary manipulation and pixel-level editing into a clear, repeatable educational activity.

Figure 1 shows the image generated by GPT to visually represent the steganographic encoding process. The base image, a 100x100 white pixel grid, was used to embed the binary message by modifying the least significant bits of each pixel.

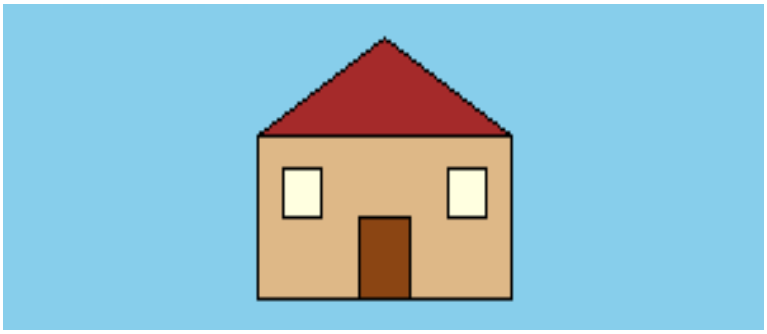

Figure 1: Steganography Image Generated

5. Future Work

We will extend this research by incorporating multimodal inputs such as images and code to support lab-based scenarios, and by evaluating AI performance on open-ended incident reports that require interpretative reasoning. Additionally, we plan to build an interactive platform where educators can test and refine their own prompts. To ensure fairness and accountability, we will introduce bias evaluation layers within the model's feedback loops.

Acknowledgment

The work supported is based upon this material by, or in part by, NSF CyberCorps(R) Scholarship for Service (SFS) 214638.

References

Chapple, M., & Seidl, D. (2024). *CompTIA Security+ Study Guide: Exam SY0-701 (9th ed.).* John Wiley & Sons, Inc.

Das, R., & Sandhane, R. (2021). *Artificial Intelligence in Cybersecurity. Journal of Physics: Conference Series, 1964(4).* https://doi.org/10.1088/1742-6596/1964/4/042072

Dokur, N. B. (2023, January 2). *Artificial Intelligence (AI) Applications in Cybersecurity.* Retrieved from https://www.researchgate.net/publication/367253331_Artificial_Intelligence_AI_Applications_in_Cyber_Security

EDUCAUSE Review. (2024). *Exploring the Opportunities and Challenges with Generative AI.* Retrieved from https://er.educause.edu/articles/2024/2/exploring-the-opportunities-and-challenges-with-generative-ai

Frontiers in Education. (2024). *Exploring the Impact of ChatGPT: Conversational AI in Education.* Retrieved from https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2024.1379796/full

Fulgencio, S.-V. (2024). *Developing Effective Educational Chatbots with GPT: Insights from a Pilot Study in a University Subject. Trends in Higher Education, 3(1),* 155–168. https://doi.org/10.3390/higheredu3010009

Khan, M. H. (2023, August 24). *Impact of Artificial Intelligence on Education.* Retrieved from https://www.wust.edu/post/impact-of-artificial-intelligence-on-education

MDPI. (2023). *Challenges and Opportunities of Generative AI for Higher Education as an Assessment Tool.* Retrieved from https://www.mdpi.com/2227-7102/13/9/856

Neurond. (2023). *Generative AI in Education: Benefits, Barriers, and Use Cases.* Retrieved from https://www.neurond.com/blog/generative-ai-in-education

OpenAI. (2023). *OpenAI for Education.* Retrieved from https://openai.com/index/introducing-chatgpt-edu/

Patel, H., & Parmar, S. (2024, March 18). *Prompt Engineering for Large Language Models.*https://doi.org/10.13140/RG.2.2.11549.93923

Ofusori, L., Bokaba, T., & Mhlongo, S. (2024). *Artificial Intelligence in Cybersecurity: A Comprehensive Review and Future Direction. Applied Artificial Intelligence, 38(1).* https://doi.org/10.1080/08839514.2024.2439609

Tajik, E. (2024). *The Ethical Concerns Surrounding GPT in Education.*https://doi.org/10.36227/techrxiv.170421413.32906559/v1

Weiler, S. C. (2024). *Developing Effective Educational Chatbots with GPT: Insights from a Pilot Study in a University Subject. Trends in Higher Education, 3(1),* 155–168. https://doi.org/10.3390/higheredu3010009