Higher Education and Practice Vol. 2 No. 12, 2025

Bridging the Practice Gap: An AI-Integrated Cybersecurity Pedagogy for Developing Operational Readiness

Li Song*, Li Jianfeng, Zeng Haowen
School of Information Engineering, Nanning College of Technology, Guilin , China
**Corresponding Author*

Abstract: As cyber threats have become increasingly sophisticated in 2025, the gap between theoretical knowledge and practical application in cybersecurity education has widened into a critical chasm. Traditional teaching methods, often reliant on static textbooks and predictable laboratory exercises, frequently produce graduates with a condition best described as "strong theory but weak hands-on skills." These students, while well-versed in concepts, struggle to apply their knowledge in dynamic, real-world scenarios, leaving them unprepared for the modern security operations center. This paper proposes a novel pedagogical framework integrated with Artificial Intelligence, termed the "AI-Cyber-Praxis" framework, designed specifically to address this skills gap. By leveraging specialized, fine-tuned Large Language Models as personalized, 24/7 pedagogical tutors and utilizing advanced AI-driven Automated Penetration Testing tools within highly dynamic virtual cyber ranges, the proposed model creates an adaptive, high-intensity, and deeply engaging learning environment. This framework shifts the traditional passive teaching mode, compelling students into active, real-time confrontation with intelligent, evolving threats. The research demonstrates that the integration of AI-enhanced feedback loops, Socratic-style guidance, and personalized challenge generation significantly improves students' critical thinking, problem-solving capabilities, and technical proficiency in core cybersecurity competencies.

Keywords: AI-Enhanced Cybersecurity Education, Large Language Models, Adaptive Cyber Ranges, Hands-On Skill Development, Automated Penetration Testing

1. Introduction

1.1 Background: The 2025 Cybersecurity Landscape

The year 2025 marks a pivotal moment in the perpetual arms race between cyber defenders and malicious actors. The landscape is no longer defined merely by the volume or velocity of attacks, but by their intelligence and autonomy. The proliferation of powerful generative AI has armed adversaries with unprecedented capabilities, leading to a surge in AI-driven polymorphic malware that alters its code with every infection to evade signature-based detection, sophisticated deepfake-based social engineering campaigns that undermine trust at a fundamental level, and automated attack frameworks that can orchestrate complex, multi-stage intrusions with minimal human intervention. This trajectory was clearly indicated by prior research, and by 2025, it has become a reality: AI-powered security testing and attack tools are not just a novelty but are on a path to outnumber human operators[1].

This new trend presents a dual challenge for cybersecurity education. Firstly, the subject matter itself has expanded. Curricula must now cover not only traditional network defense, cryptography, and penetration testing but also the security of AI systems themselves. Secondly, and more critically for the scope of this paper, the methods of teaching must evolve to match the dynamism of the threat. A student trained only on static, predictable scenarios is akin to a soldier trained for trench warfare in an era of autonomous drones.

1.2 Problem: The "Skills Gap" and Weak Practical Ability

For years, employers in the cybersecurity industry have lamented a persistent "skills gap." Graduates from top-tier academic programs often arrive with a robust theoretical understanding of security principles but lack the

practical, hands-on ability to effectively utilize industry-standard tools, respond to live incidents, or think creatively under pressure. This "strong theory, weak hands" phenomenon is a direct byproduct of traditional pedagogical models that prioritize memorization of concepts over the development of applied skills.

The deficiencies are most apparent in several key areas:

Tool Deployment and Configuration: Students may understand what a firewall or an Intrusion Detection System (IDS) does in principle, but they struggle with the complex, often frustrating, process of deploying, configuring, and troubleshooting these tools in a realistic network environment[2].

Incident Response: When faced with a simulated live-fire incident, students often freeze or follow a rigid, pre-scripted playbook that fails when the attack deviates even slightly from the textbook example. They lack the adaptive problem-solving skills to analyze unfamiliar alerts, trace attacker movements, and contain a breach in real-time.

Creative Problem-Solving: Penetration testing and ethical hacking are as much an art as a science. They require a creative, out-of-the-box mindset to chain together seemingly minor vulnerabilities into a critical exploit path. Traditional, step-by-step lab guides actively discourage this kind of thinking, instead promoting a "follow the recipe" mentality[3].

This gap between academic preparation and professional reality is no longer a mere inconvenience; in the AI-driven threat landscape of 2025, it is a critical vulnerability for our collective digital infrastructure.

1.3 Objective: An AI-Integrated "Learning by Doing" Curriculum

The central objective of the research presented in this paper is to design, implement, and evaluate a new pedagogical framework that directly confronts and remedies the problem of weak practical ability in cybersecurity education. This framework, which we have named the AI-Cyber-Praxis Model, is founded on the principle of "Learning by Doing," or more accurately, "Learning by Confronting." It seeks to transform the student from a passive recipient of information into an active participant in a dynamic, adversarial learning environment.

To achieve this, the AI-Cyber-Praxis model integrates two key streams of artificial intelligence technology directly into the fabric of the practical learning experience:

1.AI-as-Mentor: Employing highly advanced, security-specialized Large Language Models (LLMs) to serve as personalized, on-demand tutors for every student, providing guidance that is both immediate and pedagogically sound.

2.AI-as-Adversary: Leveraging the latest generation of AI-driven automated penetration testing and red teaming tools to create a dynamic, unpredictable, and adaptive cyber range environment where students can hone their skills against a non-human opponent that embodies the modern threat[4].

2. Analysis of the "Weak Practical Ability" Problem

The "strong theory, weak hands" syndrome is not a failing of the students, but rather a systemic issue rooted in the structural limitations of conventional cybersecurity education. To design an effective solution, we must first dissect the core causes of this problem. Our analysis identifies three primary contributing factors: the static nature of lab environments, the prohibitive cost of personalized guidance, and the resulting cognitive overload experienced by students.

2.1 Static and Predictable Lab Environments

The cornerstone of practical cybersecurity education has traditionally been the virtual lab environment. In its common form, this consists of a set of pre-configured virtual machines (VMs) with known vulnerabilities, accompanied by a PDF document or lab manual that guides the student through a series of steps to exploit a flaw and then patch it. While these labs serve a basic purpose in introducing concepts, they suffer from several fundamental flaws that limit their effectiveness in developing genuine practical skills.

Predictability: The environments are deterministic. The vulnerabilities are always the same, located in the same place, and exploitable by the same method. Once a student (or their predecessor from a previous semester) solves the puzzle, the solution is easily shared, reducing the exercise to a matter of mimicry rather than discovery[5]. This fails to simulate the real-world experience of a security analyst or penetration tester, whose primary task is to investigate and understand *unknown* systems

and *novel* attacks.

Lack of Dynamism: Real-world networks are not static. Services go up and down, configurations change, and users behave unpredictably. Most importantly, a real adversary reacts to the defender's actions. If a penetration tester trips an alarm, a human red teamer might change tactics, erase their tracks, or create a diversion[6]. The static VMs in a traditional lab do not react. They are passive targets, waiting to be exploited. This teaches students a dangerously simplistic view of cybersecurity as a one-sided game.

Failure to Simulate Modern Threats: Setting up a lab environment that accurately replicates a multi-stage, AI-driven attack is extraordinarily complex and time-consuming for instructors. Consequently, labs often focus on isolated, classic vulnerabilities (e.g., a single instance of an unpatched web server) and fail to represent the chained, persistent nature of modern Advanced Persistent Threats (APTs), let alone attacks orchestrated by AI agents. Students learn to find a single vulnerability, not to trace an entire attack chain across multiple systems.

2.2 The High Cost of Personalized Guidance

The second major bottleneck is the instructor-to-student ratio. In a typical university setting, a single professor or teaching assistant (TA) may be responsible for a lab session with anywhere from 30 to over 50 students. In this context, providing individualized, real-time support is a logistical impossibility.

When a student encounters a problem—be it a cryptic error message, a tool that isn't working as expected, or a conceptual misunderstanding—they are forced to raise their hand and wait. The instructor, busy with other students, may take several minutes to arrive. During this "stuck time," learning ceases, the student's focus is broken, and frustration mounts. By the time help arrives, the instructor often can only afford to provide a direct answer or a quick fix to get the student moving again. This approach sacrifices the deeper pedagogical dialogue that would empower the student to discover the solution on their own[7].

This high cost of guidance leads to several negative outcomes:

Superficial Learning: Students learn to rely on the instructor as a "debugger" rather than developing their own troubleshooting skills. They are given the fish instead of being taught how to fish.

Wasted Lab Time: A significant portion of any given lab session can be consumed by students simply waiting for help. This dramatically reduces the effective time-on-task and the amount of material that can be covered.

Disengagement: Students who are repeatedly stuck and unable to make progress become discouraged and disengaged. This is particularly damaging in a field like cybersecurity, which demands persistence and a high tolerance for failure.

2.3 Cognitive Overload and Misdirected Focus

The final piece of the puzzle is cognitive overload. The human brain has a limited capacity for working memory, and when learning a complex new skill, it is essential that this capacity is focused on the core logic of the task at hand. However, in many cybersecurity labs, students' cognitive resources are consumed by peripheral, non-essential tasks[8].

Environment Configuration: Students often spend the first hour of a lab session wrestling with virtual machine networking, installing dependencies, or debugging configuration files. This is a form of "yak shaving" that distracts from the actual security lesson. While some familiarity with system administration is necessary, it should not be the primary obstacle in a penetration testing or incident response exercise.

Tool Syntax and Esoterica: Cybersecurity tools, particularly those in the open-source world, are notoriously powerful but often have a steep learning curve with complex command-line syntax and a bewildering array of options[9]. A student trying to understand the logic of an SQL injection attack may get stuck for 20 minutes simply because they misplaced a quote in their sqlmap command. Their cognitive effort is misdirected from understanding the vulnerability to memorizing the tool's syntax.

The "Needle in a Haystack" Problem: In early stages of learning, presenting a student with a vast, un-annotated network and telling them to "find the vulnerability" can be overwhelming. They lack the established methodologies to systematically narrow the search space, leading to random, inefficient probing and immense frustration.

These three factors—static environments, insufficient guidance, and cognitive overload—

create a perfect storm that results in the "weak hands-on skills" problem. Students are trained in a way that is unrealistic, unsupported, and inefficient, leaving them unprepared for the challenges of a real-world cybersecurity career in 2025. The AI-Cyber-Praxis framework is designed to directly target and dismantle each of these obstacles.

3. Design of the AI-Enhanced Teaching Model (The "AI-Cyber-Praxis" Framework)

To overcome the systemic deficiencies outlined above, we propose the AI-Cyber-Praxis framework, a pedagogical model that fundamentally reimagines the practical cybersecurity laboratory. The framework is built upon two pillars of AI integration: an AI-Tutor for personalized mentorship and a Dynamic Cyber Range with AI Adversaries for realistic, adaptive practice. Together, these components create a symbiotic learning ecosystem where students can develop practical skills at an accelerated rate within a supportive, yet challenging, environment.

3.1 Pillar 1: AI-Driven Personalized Mentorship (The AI-Tutor)

The first pillar of the framework directly addresses the "high guidance cost" problem by providing each student with a dedicated, 24/7 AI-powered mentor. This AI-Tutor is not a simple chatbot or a search engine; it is a specialized Large Language Model designed with pedagogical principles at its core.

3.1.1 LLM Foundation and Architecture

In 2025, the capabilities of LLMs have reached a level of sophistication that makes this concept viable. Our implementation leverages a hybrid approach, using the API of a frontier commercial or open-source model for general-purpose reasoning and conversational ability. The core of the AI-Tutor, however, is a "localized security model" to ensure domain-specific knowledge and data privacy. This is achieved by taking a powerful open-source model, such as Qwen or DeepSeek—which were released in 2025 and feature advanced internal routing for complex task decomposition—and fine-tuning it on a curated dataset. This dataset includes textbooks, academic papers, security blogs, man pages and documentation for hundreds of common cybersecurity tools (e.g., Nmap, Metasploit, Wireshark), anonymized Q&A logs from previous lab sessions, a corpus of secure and insecure code snippets, and vulnerability databases like CVE descriptions.

While running a large model requires substantial GPU resources, the advent of more efficient models and optimized inference engines (like SGLang/VLLM) provides a cost-effective path for academic institutions, especially when compared to the cost of hiring numerous human TAs.

3.1.2 Core Pedagogical Functions

The AI-Tutor's value lies not in its knowledge base, but in how it imparts that knowledge. It is programmed to avoid providing direct answers and instead guide the student through a process of discovery.

Real-Time Code and Command Analysis: The AI-Tutor integrates directly with the student's lab environment (e.g., via a terminal or IDE plugin), parsing their commands and scripts as they type. If a student writes a Python exploit script with a syntax error or a logical flaw, the AI-Tutor can immediately highlight the issue and, more importantly, ask a probing question. For instance, instead of stating, "You forgot to close the socket," it might ask, "What happens to a network connection if it's not explicitly closed after use? How might that affect your script's reliability or the stability of the target service?"

Socratic Hinting for Critical Thinking: This is the core pedagogical function of the AI-Tutor. When a student encounters a difficulty—for example, reporting, “I'm running Nmap against the target IP but I don't see any open ports. I'm stuck”—the system responds with a sequence of progressive hints designed to guide inquiry. At the first level, a methodological hint is offered: “A standard Nmap scan only checks the top 1,000 most common ports. Are you certain the service you're looking for runs on one of those? What parameter could you use to check all possible ports?” If further guidance is needed, a second hint provides specific syntax: “Consider using the -p- flag in your Nmap command to perform a full port scan. Be aware, this may take longer.” Should the student remain uncertain, a third hint introduces a conceptual nudge: “Sometimes, firewalls are configured to block standard TCP scans. Have you considered trying a different scan type, like a UDP scan or a TCP SYN scan, to see if the results differ?” Through this structured Socratic dialogue, the student is prompted to activate critical thinking,

integrate discrete knowledge elements, and ultimately arrive at an independent solution. Thus, a moment of initial frustration is transformed into a meaningful learning experience, reinforcing the essential skill of methodical troubleshooting.

Conceptual Explanation on Demand: Students can ask the AI-Tutor to explain underlying concepts in context. For example, after successfully performing a cross-site scripting (XSS) attack, a student could ask, "Why did that work? Please explain the difference between stored and reflected XSS in the context of the web application I just exploited." The AI can generate a tailored explanation using the specifics of the student's recent actions as a concrete example, reinforcing the link between theory and practice.

3.2 Pillar 2: Dynamic Cyber Range with AI Adversaries

The second pillar of the framework tackles the problem of static, predictable lab environments. We replace the traditional set of fixed VMs with a dynamic, intelligent cyber range orchestrated by an AI controller. This transforms the lab from a sterile puzzle box into a living, breathing ecosystem that approximates the chaos and reactivity of a real-world network.

3.2.1. AI-Orchestrated Environment Generation

The cyber range is not a single, fixed topology. At the beginning of a lab module, an AI orchestrator procedurally generates a unique network environment for each student or small group. This includes:

Variable Topologies: Generating diverse network layouts with different subnets, DMZs, and routing rules.

Dynamic Service Deployment: Populating the network with a mix of servers and workstations running different operating systems and services.

Programmatic Vulnerability Insertion: The orchestrator draws from a large library of vulnerabilities (misconfigurations, unpatched software, weak credentials, custom-coded flaws) and intelligently seeds them throughout the network. It can create complex attack chains where exploiting a low-privilege web server is merely the first step to reaching a critical database on a separate, firewalled segment. This ensures that no two students have the exact same problem set, eliminating simple answer-sharing and forcing original analysis.

3.2.2 Automated Red Teaming with AI Adversaries

The most significant innovation of this cyber range is the introduction of an AI-driven adversary. Instead of being a passive target, the network is actively inhabited by an automated red team agent powered by the same class of technologies used in state-of-the-art automated penetration testing. As of 2025, platforms like PentestGPT and the newer, multi-agent xOffense framework have demonstrated significant capabilities on industry benchmarks. By integrating an educational variant of these tools, the AI Adversary can perform a variety of actions in real-time:

Simulate Diverse Attack Patterns: The AI Adversary can autonomously execute attack paths against the student's "blue team" infrastructure or other targets. It can perform reconnaissance, attempt SQL injections, escalate privileges, and move laterally, providing a constant stream of realistic, malicious traffic for students practicing defensive skills.

Active Counter-Reconnaissance: When a student performs a port scan, the AI Adversary can detect it and respond by feeding the student false information (e.g., opening decoy "honeypots"), blocking the student's IP, or even launching a counter-attack against the student's analysis machine. This teaches the critical lesson that attackers are not passive and that operational security matters.

Real-time Defense: When a student is on the offensive, the AI Adversary can play the role of a rudimentary "blue team." If a student gains a foothold by exploiting a service, the AI might detect the anomalous process and terminate it, forcing the student to find more persistent or stealthy methods.

3.2.3 Adaptive Difficulty and Biometric Feedback

To address cognitive overload and maintain optimal engagement, the AI orchestrator modulates the difficulty of the challenges in real-time based on the student's performance and physiological state. This creates a personalized learning curve, a state of "flow" where the challenge is perfectly matched to the student's ability.

The system integrates data from three primary sources to form a comprehensive learning profile. Performance metrics, including task success/failure rates, completion time, and the

number of hints requested from the AI-Tutor, provide a direct measure of proficiency and engagement. These are supplemented by behavioral metrics, such as keyboard dynamics (typing speed, error frequency, and backspace usage), command execution patterns, and periods of inactivity indicative of "stuck time." Optionally, and strictly with user consent, biometric data may be incorporated; utilizing a standard webcam and computer vision algorithms, the system can derive heart rate variability (HRV) via facial photoplethysmography, which detects subtle shifts in skin coloration. Elevated or irregular HRV may signal states of stress or frustration, whereas consistently low variability could suggest disengagement or boredom, thereby offering an additional, physiological dimension to the assessment of learner state.

Based on this multi-modal input, the AI orchestrator can make real-time adjustments. If a student is succeeding too easily, it can deploy a new, unpatched server or have the AI Adversary launch a more complex attack. Conversely, if a student is clearly frustrated (high HRV, long inactivity), the system can proactively offer a hint through the AI-Tutor or temporarily disable a defensive mechanism to allow the student to make progress. This multi-modal approach, while novel in the cybersecurity domain, is grounded in established principles from adaptive learning and intelligent tutoring systems research.

4. Implementation Strategies

The successful deployment of the AI-Cyber-Praxis framework requires more than just advanced technology; it necessitates a thoughtful integration into the curriculum and a re-architecting of the typical lab workflow. This section outlines a practical strategy for bringing the framework to life within a university course structure.

4.1 Curriculum Integration: The "AI + X" Modular Approach

We propose structuring the practical components of cybersecurity courses around an "AI + X" model, where 'X' represents a core security domain (e.g., Defense, Offense, Forensics). The AI-Cyber-Praxis framework serves as the underlying platform for all these modules, providing a consistent yet flexible environment.

4.1.1 Module 1: AI for Defense (Blue Teaming)

This module equips students with the skills to build and operate AI-powered defensive tools. The core project involves developing a custom Network Intrusion Detection System (NIDS). The process begins with students using standard machine-learning libraries like TensorFlow or PyTorch to design a model (e.g., an autoencoder) that learns the "normal" traffic patterns of their assigned cyber range. Throughout this phase, the AI-Tutor provides support by debugging Python code, clarifying concepts like feature engineering for network packets, and explaining performance metrics. Once the NIDS is deployed, the AI orchestrator activates an AI Adversary to execute a progression of attacks, from simple port scans to stealthy data exfiltration. The student's task is to monitor their AI-driven NIDS, distinguish true-positive alerts from false positives, and respond accordingly. This dynamic simulation underscores the practical challenges of AI security tools, reinforcing the importance of model tuning and alert validation in operational environments.

4.1.2 Module 2: Attacking AI (Red Teaming against AI)

This module shifts students into an offensive role, teaching them to assess and compromise the security of AI models themselves, a field known as adversarial machine learning. Building on the previous module, the primary objective is to attack the NIDS they or their peers developed. Students learn the principles of evasion attacks (crafting malicious packets misclassified as benign) and data poisoning attacks (injecting samples into training data to create backdoors). The AI-Tutor assists with the complex mathematics and programming required to generate adversarial examples using techniques like the Fast Gradient Sign Method, offering code templates and clarifying the underlying theory. The dynamic cyber range provides a safe, sandboxed environment for execution, with the AI orchestrator hosting additional vulnerable AI models (e.g., image classifiers) as targets to broaden student experience. This module prepares students to defend the next generation of AI-driven systems.

4.2 The "Human-AI" Collaborative Lab Workflow

The AI-Cyber-Praxis model redefines the

structure of a lab session, shifting it from a linear, step-by-step process to a cyclical, three-phase workflow: Preparation, Execution, and Reflection. This workflow maximizes the collaborative potential between the student and the AI systems.

4.2.1 Phase A: Preparation and Reconnaissance

This phase focuses on planning and offloading tedious tasks to the AI, allowing the student to concentrate on high-level strategy. Given a mission objective (e.g., gain administrative access to a specific server), students collaborate with the AI-Tutor, which serves a multifunctional role. First, as a strategist, it suggests structured workflows like host discovery followed by port scanning and vulnerability research. Second, as an automation assistant, it helps students generate Python scripts to parse Nmap results and query exploit databases, imparting practical SecOps skills. Third, as a visualization tool, it dynamically generates and updates live network topology diagrams based on discoveries, providing an intuitive overview of the attack surface. This integrated workflow cultivates essential competencies in systematic reconnaissance, automation, and situational awareness.

4.2.2 Phase B: Execution and Confrontation

This is the core hands-on phase where the student attempts to achieve their objective, engaging directly with the target systems and the AI Adversary. As the student performs attacks and exploits vulnerabilities, AI collaboration plays a dual role. First, the AI-Tutor acts as an in-context copilot, delivering real-time support through Socratic hinting to guide the student past technical obstacles without revealing direct solutions, while also acting as a safety net by warning against potentially destructive commands (e.g., prompting for confirmation before rm -rf /). Second, the AI Adversary functions as an active sparring partner, dynamically responding by patching vulnerabilities, detecting intrusions, and initiating countermeasures to expel the student from compromised systems. This real-time "cat-and-mouse" interaction fosters persistence, stealth, and adaptive problem-solving skills that cannot be replicated in static setups.

4.2.3 Phase C: Reflection and Reporting

After the lab session concludes, the focus shifts to learning from the experience, a phase heavily automated by the AI to provide deep, objective feedback. The student first composes a brief summary of their actions and takeaways. In parallel, the AI orchestrator generates a comprehensive, personalized post-mortem report. This report includes: (1) an attack-path visualization that graphically traces every command and system accessed; (2) a comparison of the student's path with an AI-derived optimal sequence (e.g., from a system like xOffense), highlighting missed opportunities or more efficient exploits; (3) performance analytics such as time-to-first-compromise, total duration, and failed-command counts; and (4) an annotated event log that links student actions to adversary responses (e.g., showing that a specific Nmap scan triggered a new firewall rule). This data-rich debrief directly illustrates cause-and-effect dynamics, reinforcing operational insights and fostering metacognitive awareness.

5. Ethical and Technical Challenges

The implementation of a sophisticated framework like AI-Cyber-Praxis is not without its challenges. A responsible and critical approach requires acknowledging and proactively addressing several significant ethical, pedagogical, and technical hurdles.

5.1 Over-Reliance on AI and the Atrophy of Foundational Skills

A primary pedagogical concern is the risk of student over-reliance on AI tools. If the AI-Tutor is too helpful, students may successfully complete tasks without developing a fundamental understanding of the underlying principles. This could lead to a new form of "weak hands," where students are proficient at using their AI copilot but are lost without it.

Mitigation Strategy: The design of the AI-Tutor, with its emphasis on Socratic hinting over direct answers, is the primary defense. The system is intentionally designed to be a "Socratic sparring partner," not a solution oracle. Furthermore, the curriculum must include assessments conducted without AI assistance to ensure students are internalizing core skills. The AI-generated post-mortem reports also play a role by forcing students to reflect on the logic of the optimal path, not just the commands themselves.

5.2 Data Privacy and Student Monitoring

The AI-Cyber-Praxis framework inherently collects a vast amount of data on student performance, including every command typed, every error made, and potentially biometric data. This raises significant privacy concerns that must be managed in compliance with regulations such as the EU's General Data Protection Regulation and the California Consumer Privacy Act.

Mitigation Strategy: A multi-layered privacy-preserving approach is essential.

1.Transparency and Consent: Students must be fully informed at the beginning of the course about what data is being collected, how it is being used to personalize their learning, and who has access to it. Explicit, opt-in consent must be obtained, particularly for any biometric data collection.

2.Anonymization and Aggregation: All data used for research or model improvement must be rigorously anonymized to remove personally identifiable information.

3.On-Premise Deployment: To maintain full data control, the localized security LLM and the cyber range orchestrator should be hosted on the university's internal servers rather than relying solely on third-party cloud services for sensitive processing.

4.Privacy-Preserving Techniques: For fine-tuning the AI-Tutor model on student data, techniques like Differential Privacy (DP) can be employed. DP adds statistical noise during training, providing a mathematical guarantee that the model does not memorize or reveal information about any single individual.

5.3 Technical and Financial Viability

The implementation and sustained operation of such a framework entail a considerable and ongoing technical and financial investment for academic institutions. Infrastructure costs are substantial, primarily driven by the GPU hardware required to train, fine-tune, and run multiple instances of large language models, as well as to simulate complex network environments; these costs encompass not only initial capital expenditure but also recurring expenses for power and cooling. Furthermore, the development and maintenance of the system itself represent a major software engineering undertaking, involving the creation of the AI orchestrator, the pedagogical logic for the AI-Tutor, and an extensive library of vulnerabilities and network components. This software base requires continuous updates to integrate new security tools, patch its own potential vulnerabilities, and remain aligned with the evolving threat landscape. An additional, inherent challenge is posed by model drift and potential jailbreaking: large language models may exhibit performance drift over time, while resourceful students may attempt to bypass the AI-Tutor's Socratic guidance to obtain direct answers. Similarly, the difficulty level of the AI Adversary may drift, becoming disproportionately easy or challenging. Consequently, the system demands persistent monitoring, periodic reevaluation, and careful fine-tuning to preserve its intended pedagogical rigor and effectiveness.

Mitigation Strategy: Costs can be mitigated through a combination of using open-source models, leveraging more efficient inference techniques, and seeking university investment by framing the framework as a critical infrastructure for producing top-tier, job-ready graduates. The development effort can be a project for graduate students or a university's IT research division, creating a synergistic research and teaching opportunity.

6. Conclusion

The year 2025 is characterized by an escalating cyber threat landscape where the advantage increasingly belongs to those who can operate at the speed and scale of artificial intelligence. In this environment, traditional cybersecurity education, with its static labs and passive learning models, is no longer sufficient. The "strong theory, weak hands" problem has evolved from a minor skills gap into a critical strategic disadvantage.

This paper has proposed the AI-Cyber-Praxis framework, a new pedagogical model for practical cybersecurity education designed for the modern era. By integrating AI as both a personalized mentor and an intelligent adversary, the framework directly dismantles the core obstacles to effective skill acquisition. The AI-Tutor solves the problem of scale and guidance cost, providing every student with immediate, Socratic-style support that fosters critical thinking. The Dynamic Cyber Range with AI Adversaries, powered by the latest automated red teaming technologies, solves the problem of unrealistic lab environments, immersing students in a reactive and challenging world that mirrors real-world

confrontations.

We have acknowledged the significant ethical and technical challenges, from the risk of AI over-reliance to the critical importance of student data privacy and the substantial infrastructure investment required. These are not trivial concerns, but with careful design—emphasizing Socratic methods, robust privacy-preserving techniques like differential privacy, and strategic institutional planning—they are surmountable.

The integration of AI into cybersecurity education is no longer an option, but a necessity. By shifting the pedagogical focus from the passive consumption of theoretical knowledge to a model of AI-supported active confrontation, we can effectively bridge the chasm between classroom learning and the demanding requirements of the professional cybersecurity world. The AI-Cyber-Praxis framework offers a viable, effective, and forward-looking path to cultivating the next generation of cyber defenders—professionals equipped not just with knowledge, but with the battle-tested, adaptive skills needed to secure our future in an increasingly intelligent and automated world.

Acknowledge

This article is a phased research outcome of the 2025 Guangxi Higher Education Undergraduate Teaching Reform Project titled “Reform and Practice of Teaching Model for Cybersecurity Technology Courses Based on Artificial Intelligence+ under the Background of New Engineering” (No. 2025JGB549).

References

[1] Lingzhi Wang, Xinyi Shi, Ziyu Li, et al, “Automated Penetration Testing with LLM Agents and Classical Planning”, 2025, arXiv:2512.11143.

[2] Gholami, Yazi . "Large Language Models (LLMs) for Cybersecurity: A Systematic Review." World Journal of Advanced Engineering Technology and Sciences 13.1(2024):057-069.

[3] Lei Wu, Xiaofeng Zhong, Jingju Liu, and Xiang Wang. PTGroup: An Automated Penetration Testing Framework Using LLMs and Multiple Prompt Chains. In Advanced Intelligent Computing Technology and Applications: 20th International Conference, ICIC 2024, Tianjin, China, August 5–8, 2024, Proceedings, Part IX. Springer-Verlag, Berlin, Heidelberg, 220–232.

[4] Luis Javier García Villalba. "Analysis of Autonomous Penetration Testing Through Reinforcement Learning and Recommender Systems." Sensors 25(2025).

[5] Deng, Gelei, et al. "{PentestGPT}: Evaluating and harnessing large language models for automated penetration testing." 33rd USENIX Security Symposium (USENIX Security 24). 2024.

[6] Hannay, Jo Erskine, Audun Stolpe, and Muhammad Mudassar Yamin. "Toward ai-based scenario management for cyber range training." International Conference on Human-Computer Interaction. Cham: Springer International Publishing, 2021.

[7] Umaroh, Liya, Mukaromah Mukaromah, and Muhammad Naufal. "Enhancing speaking skill through ai-powered technology." Seminar Nasional Teknologi Dan Multidisiplin Ilmu (SEMNASTEKMU). Vol. 3. No. 1. 2023.

[8] Andreolini, Mauro, et al. "A framework for the evaluation of trainee performance in cyber range exercises." Mobile Networks and Applications 25.1 (2020): 236-247.

[9] Chopra S, Ahmad H, Goel D, et al. Chatnvd: Advancing cybersecurity vulnerability assessment with large language models[J]. arXiv preprint arXiv:2412.04756, 2024

.