Malware Analysis Education Meets LLMs: Understanding Student Use of LLMs in Malware Analysis Education

Orçun Çetin* and Nazlı Bıyıklı

Faculty of Engineering and Natural Sciences, Sabancı University, Istanbul, Turkey
Email: orcun.cetin@sabanciuniv.edu (O.C.); nazlibiyikli@sabanciuniv.edu (N.B.)
*Corresponding author

Manuscript received August 14, 2025; revised September 1, 2025; accepted November 6, 2025; published March 17, 2026

***Abstract*—Large Language Models (LLMs) are increasingly used in cybersecurity, both as practical tools in real-world tasks like penetration testing and reverse engineering, and as educational aids for students learning complex analysis techniques. While recent research highlights their potential to automate code analysis, deobfuscation, and threat detection, less is known about how students actually use these models during malware analysis courses. To address this gap, we conducted a survey of 37 students enrolled in university-level malware analysis courses. Our findings show that all participants reported using LLMs, primarily for assignments and labs (70.2%) and to better understand course content (59.4%). Students primarily analyzed outputs from Interactive Disassembler Pro (IDA Pro) (83.7%), followed by OllyDbg and Wireshark (43.2%). They mainly used LLMs for advanced static analysis, especially for disassembled code interpretation (37.8%). While 59.4% of students reported no major issues when using LLMs, 27% encountered refusals to respond, primarily due to ethical safeguards built into the models, and others noted inaccurate or overly generic responses and token-size limits. In terms of satisfaction, 67.5% of students reported positive experiences with LLMs, and 81% indicated they were likely to continue using them for cybersecurity-related tasks in the future. These findings suggest the need for responsible integration of LLMs into cybersecurity education through lecturer guidance, ethical transparency, and effective assessment design. Overall, they highlight both the strengths and limitations of LLMs in supporting advanced technical learning.**

***Keywords*—malware analysis, large language model, education, Artificial Intelligence (AI) in cyber security, generative AI**

I. Introduction

In recent years, the emergence of LLMs has significantly impacted the field of cybersecurity, particularly in how security professionals analyze code, detect threats, and interpret complex malware behavior. Models such as ChatGPT, Claude, and Llama are being increasingly used to automate tasks. LLMs are being integrated into workflows that traditionally relied on manual analysis. Recent research has shown their potential to assist with phishing generation and detection, vulnerability detection, reverse engineering, code deobfuscation, and penetration testing [1–6]. These developments demonstrate the growing role of LLMs as support tools across both offensive and defensive security tasks. Beyond cybersecurity applications, LLMs have also gained significance in educational settings, particularly in higher education. Students increasingly turn to LLMs as virtual assistants to help understand course material, generate summaries, receive feedback, and clarify complex concepts. Studies show that AI-driven tools like ChatGPT can enhance students' motivation, confidence, and perceived learning outcomes, especially in technical disciplines such as computer science [7]. Also, LLMs can improve academic performance, self-efficacy, and reduce learning anxiety among university students [8]. Among the most frequent use cases of LLMs in academic settings were completing assignments and retrieving information [9]. This suggests that LLMs are not only being used by cybersecurity professionals for real-world tasks, but are also shaping how students learn and interact with complex technical content in academic settings. While prior studies have explored the use of LLMs in either cybersecurity workflows or general academic contexts, much less is known about how students engage with LLMs in highly technical, security-focused courses that blend both domains. Malware analysis presents a unique challenge: it demands both theoretical understanding and hands-on practice with real-world tools and techniques as well as dangerous malicious software. These tasks often involve interpreting low-level code, recognizing obfuscation patterns, and correlating system-level activities. Despite this complexity, there is limited empirical research on how students in such courses actually make use of LLMs. To address this gap, we conducted a survey of 37 students enrolled in malware analysis courses, aiming to investigate how they used LLMs throughout different stages of the course and for which specific tasks. The questionnaire covered a range of topics, including the types of analyses where students found LLMs most and least helpful, the tools whose outputs were interpreted with LLM support, their perceived helpfulness of LLMs for answering course-related questions, interpreting disassembled code and their motivations for using these models, such as deepening their understanding of course concepts or revisiting examples discussed in class. In addition, students were asked about their overall satisfaction with LLM usage, challenges they encountered, and recommendations for future improvement.

This study is guided by four research questions. First, we investigate how students employ LLMs to support malware analysis tasks in a university-level course. Second, we explore which specific tool outputs, and tasks students find LLMs most and least helpful. Third, we examine the challenges and limitations students encounter when using LLMs for malware analysis education. Finally, we explore the broader implications of LLM adoption for student learning and the future of cybersecurity training.

This paper makes the following key contributions:

- We present the first empirical study that systematically investigates how students use LLMs during a university-level malware analysis course.
- Our findings reveal that all students enrolled in the

malware analysis course utilized LLMs, with ChatGPT emerging as the overwhelmingly preferred choice among participants. Alternative models such as Claude (13.5%), Grok (8.1%), and Gemini/Bard (8.1%) were used by only a small subset of students.

- The primary reasons for using LLMs included seeking assistance with assignments and lab work (70.2%) and gaining a better understanding of course concepts (59.4%).
- When examining tool usage, students most frequently analyzed the output of IDA Pro (83.7%) with the help of LLMs, followed by OllyDbg and Wireshark (both at 43.2%).
- Overall, 67.5% of students reported positive experiences with LLMs, and a strong majority (81%) indicated that they are likely to continue using LLMs for cybersecurity-related tasks in the future.

II. Literature Review

It is undeniable that LLMs have significantly integrated into education and are widely adopted by students. Large-scale surveys confirm that students mainly adopt these tools for assignments, research, and exam preparation. For example, Wang and Li's study of 721 university students found that the primary purposes of LLM use were for completing assignments (63.11%) and searching for information (53.08%) [9], while Boubker's survey of Moroccan higher education students introduced uses such as preparing presentations and studying for exams [10]. Several studies also suggest that LLMs boost students' confidence and engagement in technical courses. Amoozadeh *et al*. [7] surveying 253 students enrolled in computer science courses, reported that greater trust in generative AI tools was associated with students having higher levels of confidence and motivation in programming. Wu and Yu's meta-analysis similarly shows that using LLMs can positively influence dimensions in students' learning outcomes, such as academic performance, self-efficacy, motivation, interest, perceived learning value, and anxiety reduction [8]. Hanifi *et al*. [11] extend these findings in a software engineering context, showing that over 93% of students used ChatGPT in projects, with many reporting improved productivity and a strong intention to continue using the tool in future work.

Experimental studies further highlight the potential of LLMs as teaching assistants. Essel *et al*. studied 68 undergraduate students in a multimedia programming course and found that those using a teaching assistant chatbot achieved significantly better academic performance, particularly in contexts with high student-teacher ratios such as in Ghana [12]. Similarly, Chen *et al*. [13] argue that LLMs can support students in large classes or when they feel hesitant to approach instructors, helping them remain more involved in the course. Viorennita *et al*. [14] also emphasize that ChatGPT can offer feedback on assignments and provide learning resources tailored to students' specific needs.

While the studies mentioned so far highlight the benefits of integrating LLMs in education, recent research also raises concerns about potential drawbacks and limitations associated with their widespread use, with academic ethics being one of the most prominent issues. This concern is supported by Maulana *et al*. who emphasize that while ChatGPT can help students complete academic tasks more efficiently, its overuse may result in violation of academic ethics and weaken students' ability to think critically and creatively [15]. Quintans-Júnior *et al*. [16] suggest that ChatGPT solely relies on existing data, and lacks the human capacity for analytical reasoning, such as making informed technical and scientific judgements. In a broader view, Guilherme argues that the increasing reliance on educational technologies can harm the quality of teacher-student relationships, arguing that education, while is about learning skills, also contributes to a more personal level, character formation [17]. They also suggest that trends like learnification reduce teachers to facilitators, neglecting the deeper human and ethical role of education. Similarly, Nguyen emphasizes that LLMs cannot replicate the emotional and social dimensions of teachers, who can adjust their methods to individual student needs, interpret nonverbal cues, offer encouragement, and foster a supportive learning environment [18].

While there is extensive research on LLM use in education and their effects on students, most of these studies take a broad, general view of educational contexts, rather than focusing on specialized domains. However, there remains a clear gap in student-level research within advanced cybersecurity courses, where learning contexts are highly specialized and technically demanding, such as penetration testing, malware analysis, incident response, cryptography, or reverse engineering. In contrast, most domain-specific insights come from industry settings, focusing on professional analysts and operational contexts. For instance, across multiple studies, LLMs have shown strong potential to support digital forensic workflows by accelerating evidence extraction, automating scripting tasks, and generating narrative analysis, often achieving accuracy levels comparable to human analysts while significantly reducing analysis time [19]. Similarly, in secure coding, LLMs have demonstrated strong performance in identifying and fixing common vulnerabilities, however their effectiveness declines for more complex or context-specific issues [20]. In DevSecOps, AI-driven tools, such as Microsoft's Security Copilot, are increasingly used to support early security activities like threat modeling and impact analysis, and trials show that experienced analysts completed tasks 22% faster and 7% more accurately when using Copilot [21]. These examples show that while LLMs are increasingly used in specialized cybersecurity workflows in industry, comparable student-level research is lacking. This paper addresses this gap by focusing on how university students use LLMs in their malware analysis course.

III. Materials and Methods

A. Study Setup

To conduct this survey, we invited students from Sabancı University who were enrolled in the Malware Analysis and Detection course. The course is offered under three different codes: SEC 503, SEC 530, and CS 48008, corresponding to different academic tracks. SEC 503 is designed for non-thesis master's students, SEC 530 is offered to thesis-track master's students, and CS 48008 is taken by undergraduate students, typically in their final year. The sample was limited to those who took the course during the 2023–2024 and 2024–2025

academic years. In total, 61 students participated in at least one offering of the course during this period. From these students only 37 of them participated in our study. While the course codes differ based on academic level, the content is largely consistent across all sections. The curriculum covers a broad range of topics, including malware behavior, basic and advanced static and dynamic analysis, sandbox environments, malicious document analysis, YARA rule creation, Windows API usage, disassembly fundamentals, packers, obfuscated malware, anti-VM techniques, and analyzing malware generated in Java and .NET languages. Participation in the survey was entirely voluntary, and students were informed of their right to decline or withdraw from the study at any point without any penalties or loss of academic benefits. To ensure the privacy and ethical integrity of our research, we emphasize that all participation in this study is strictly confidential. Individual responses have been anonymized and are reported only in aggregate form, making it impossible to identify any specific participant. This approach allows us to gather meaningful insights while fully respecting the privacy of each contributor. Participants were informed of these measures prior to their involvement, in line with standard research ethics practices. All of this information was clearly outlined in the consent form, which participants were required to read and agree to before proceeding with the survey. In addition, this study has been reviewed and approved by the Ethics Committee at Sabancı University, ensuring compliance with established ethical standards for research involving human subjects. In designing our survey, we followed a structured, two-part approach to collect both background information and detailed insights into students' LLM usage. The first section focused on demographic and educational context, collecting information on participants' age, gender, field of study, academic standing, and current involvement in cybersecurity through work or internships. The second section investigated students' use of LLMs specifically within the malware analysis course. The survey questions were designed to evaluate which malware analysis techniques students most and least used with LLMs, the tools whose outputs were interpreted with LLM assistance, and the overall frequency and purpose of LLM use, such as for assignments, exam preparation, or understanding course concepts. We used multiple-choice formats for questions on tool usage, LLM purpose, and frequency; Likert scales for evaluating helpfulness and satisfaction; and single-choice formats for demographic questions to ensure clarity and consistency. The sequence of questions was carefully ordered to ensure logical flow and contextual relevance. For instance, questions about tool-specific LLM usage were followed by questions on students' preferences for integrating LLM support into existing analysis tools. Finally, open-ended questions were included to gather qualitative insights on challenges students faced and feature recommendations, offering students an opportunity to elaborate on their experiences in their own words. Before distributing the final survey, we conducted a small pilot test with 5 participants to evaluate question clarity, logical flow, and completion time. Feedback from this pilot was used to refine the wording and structure of several items, ensuring that the final survey was clear, coherent, and reliable.

B. Participants' Profile

The study involved a total of 37 participants, with a diverse distribution across age and employment status, shown in Table 1 and Table 2. The age breakdown showed that the majority were between 18–24 and 25–34 years old, comprising 17 and 16 participants respectively. A smaller number of participants were in the 35–44 and 45 or older categories, with two individuals in each. In terms of gender, the sample was predominantly male, with 28 male and 9 female participants. Employment status varied, with most participants (22) not currently working or doing an internship. Thirteen participants reported working full-time, while one participant each was engaged in part-time work or an internship. Moreover, Table 3 shows study participants' academic levels. The largest groups were MSc students in the non-thesis program and undergraduate students, each consisting of 12 participants. This was followed by MSc students (thesis-based), with 11 participants, indicating a relatively balanced distribution between postgraduate and undergraduate participants in the study. The majority of participants in the study were from Computer Science and Engineering, comprising 23 individuals, which reflects the study's primary focus on technical domains shown in Table 4. Cybersecurity was represented by 2 participants, while 3 participants reported Statistics as their major. Additionally, 8 participants came from various other engineering disciplines, and 1 participant had a background in Political Science, indicating a small but diverse academic representation.

Table 1. Age distribution

Age Group	Number of Participants
18–24	17 (45.95%)
25–34	16 (43.24%)
35–44	2 (5.41%)
45 or older	2 (5.41%)

Table 2. Work distribution

Employment Status	Number of Participants
Not working	22 (59.46%)
Full-time working	13 (35.14%)
Doing internship	1 (2.7%)
Part-time working	1 (2.7%)

Table 3. Academic standing

Grade Level	Number of Participants
Undergraduate student	12 (32.43%)
MSc student without thesis	12 (32.43%)
MSc student	11 (29.73%)
Graduate	1 (2.7%)
MSc graduate	1 (2.7%)

Table 4. Major or area of study distribution

Field of Study	Number of Participants
Computer Science and Engineering	23 (62.16%)
Statistics	3 (8.11%)
Cybersecurity	2 (5.41%)
Political Science	1 (2.7%)
Other Engineering	8 (21.62%)

IV. Result and Discussion

A. LLM Usage in the Malware Analysis Course

To understand students' LLM usage in malware analysis course, we started off our research by asking if they utilized any LLMs during the course, and if so, which models they used. Table 5 shows the distribution of LLMs used by

students during the malware analysis course, showing that ChatGPT was by far the most used model, with all of the students reporting its use. This result is unsurprising, given ChatGPT's widespread popularity and accessibility compared to other LLMs. Alternative models like Claude (13.5%), Grok (8.1%), and Gemini/Bard (8.1%) were used by a smaller subset of students, indicating occasional experimentation or use as supplementary tools alongside ChatGPT. A few students also explored niche tools like Cursor, DeepSeek, and Llama 2 and 3, each used by only one student (2.7%). It should also be noted that in this question, participants were allowed to select multiple models, so the total number of responses exceeds the number of participants (37).

Table 5. Distribution of LLMs used by students during the malware analysis course

LLMs Used	Number of Participants
ChatGPT	37 (100.00%)
Claude	5 (13.51%)
Grok	3 (8.11%)
Gemini / Bard	3 (8.11%)
Cursor	1 (2.70%)
DeepSeek	1 (2.70%)
Llama 2/3	1 (2.70%)
None	0 (0.00%)

To further investigate how LLM usage varied across different stages of malware analysis course, we asked students to indicate in which type of analysis they used LLMs the most and the least (see Table 6 and Table 7). Basic static analysis and advanced static analysis emerged as the two most frequently reported analysis types where students relied on LLMs, while advanced dynamic analysis was the least. One likely reason for this pattern is that static analysis tasks do not require executing the malware and are generally more straightforward to represent in text form, making them easier to query with LLMs. This likely influences students' preference for seeking assistance from LLMs during these stages. On the other hand, students reported using LLMs in advanced dynamic analysis the least. This may be due to the higher complexity of these types of tasks, involving execution of the malware in a controlled environment, monitoring the runtime behavior and analyzing real-time system interactions. Also, transferring disassembly code from debuggers like OllyDbg into LLM input formats can be more challenging, potentially limiting students' ability to effectively utilize LLMs during this stage of analysis. This suggests that current LLMs may be more helpful in guiding students through the interpretation of disassembled code and static features, particularly for beginners, than for supporting advanced dynamic analysis.

Table 6. Types of malware analysis where students reported using LLMs the most

Student Response	Number of Participants
Basic Static Analysis	12 (32.43%)
Basic Dynamic Analysis	2 (5.41%)
Advanced Static Analysis	14 (37.84%)
Advanced Dynamic Analysis	5 (13.51%)
Not sure	4 (10.81%)

When comparing subgroups, almost all undergraduate students, 12 out of 13 (92.3%), reported using LLMs primarily for static analysis, where MSc students' answers were more scattered. Out of 24, 14 students answered using LLMs for static analysis (58.3%), 6 for dynamic (25%), and 4 were unsure. This distribution suggests that while undergraduate students tend to align their LLM use with analysis types that are more easily queried in text (i.e., static analysis).

Table 7. Types of malware analysis where students reported using LLMs the least

Student Response	Number of Participants
Basic Static Analysis	4 (10.81%)
Basic Dynamic Analysis	3 (8.11%)
Advanced Static Analysis	3 (8.11%)
Advanced Dynamic Analysis	17 (45.95%)
Not sure	10 (27.03%)

To gain a more detailed view of students' LLM usage, we asked which tools' outputs they had analyzed using LLMs, which can be seen in Table 8, IDA Pro, an advanced static analysis tool emerged as the most commonly analyzed, where over 83.7% of the participants selected it. This shows that students frequently used LLMs to interpret disassembled code from IDA Pro. Interestingly, OllyDbg and Wireshark were among the most frequently selected tools in this question, each receiving 16 votes (43.2%), which does not directly align with earlier results which indicated that LLMs were primarily used for static analysis tasks. One possibility of this discrepancy may be because these tools were explicitly required in assignments or labs, leading to more frequent use regardless of students' overall preferences. Another explanation could be that the tools' interfaces or outputs can be harder to understand, leading students to utilize LLMs more for a better interpretation. While the previous question captured tools students had already used with LLMs, we also asked their opinions of where LLM integration would be the most beneficial. Specifically, we asked which tool they would like to have integrated with an LLM to better understand its output, which is shown in Table 9. Consistent with our previous question, IDA Pro was the most selected tool, chosen by 78.3% of participants. This was followed by OllyDbg with 35.1%. Wireshark and Process Monitor (Procmon) were also next in line, each selected by 24.3% of students, indicating that these tools are also perceived as challenging or in need of interpretation support. This alignment between the actual usage and the desired integration shows that students tend to seek LLM assistance in tools that are more complex, giving out low-level outputs. Specifically, using IDA Pro and OllyDbg requires technical knowledge to interpret disassembled code, which may explain why students both use LLMs with them and also would like an integration with LLMs. Similarly, Wireshark and Process Monitor produce large volumes of logs, where LLMs could help filter, summarize, or explain specific patterns or anomalies. Interestingly, while Ghidra received the fewest selections in the previous question, where students were asked which tool outputs they currently analyze using LLMs, it ranked significantly higher when students were asked whether they would like to see LLM integration within the tool. This contrast suggests that although Ghidra may not have been frequently used in conjunction with LLMs during the course, students recognize its potential and express a strong interest in enhanced LLM support for future use. To assess students' perceptions of

LLM usefulness in understanding low-level code, we asked them to rate how helpful the LLM was in interpreting disassembled code, such as the disassembly displayed in IDA Pro. The results, shown in Table 10, indicate that overall perceptions were highly positive. The majority of the students rated the LLM as either "Helpful" or "Extremely helpful", together making up 70.2% of all responses. These high ratings suggest that students found LLMs helpful in making complex disassembled code easier to understand.

Table 8. Tools whose outputs were analyzed by students using an LLM

Student Response	Number of Participants
IDA Pro	31 (83.78%)
OllyDbg	16 (43.24%)
Wireshark	16 (43.24%)
PEiD	12 (32.43%)
Process Monitor (Procmon)	12 (32.43%)
PEStudio	10 (27.03%)
Detect It Easy (Die)	8 (21.62%)
CFF Explorer	7 (18.92%)
Regshot	7 (18.92%)
YARA & Malicious Documents Analysis Tools	6 (16.22%)
Process Explorer	5 (13.51%)
Bytecode Viewer	5 (13.51%)
dnSpy	4 (10.81%)
FakeNet	3 (8.11%)
Ghidra	2 (5.41%)
None	0 (0.00%)

Table 9. Tools students would most like to see integrated with an LLM

Student Response	Number of Participants
IDA Pro	29 (78.38%)
OllyDbg	13 (35.14%)
Wireshark	9 (24.32%)
Process Monitor (Procmon)	9 (24.32%)
PEStudio	8 (21.62%)
Ghidra	8 (21.62%)
Process Explorer	6 (16.22%)
YARA & Malicious Documents Analysis Tools	6 (16.22%)
CFF Explorer	4 (10.81%)
PEiD	4 (10.81%)
FakeNet	4 (10.81%)
Regshot	4 (10.81%)
Bytecode Viewer	4 (10.81%)
Detect It Easy (Die)	2 (5.41%)
dnSpy	2 (5.41%)
None	3 (8.11%)

Table 10. Student responses of how helpful LLMs were in terms of understanding or explaining the behavior of disassembled code

Student Response	Number of Participants
Not helpful at all	0 (0.00%)
Slightly helpful	2 (5.41%)
Somewhat helpful	7 (18.92%)
Helpful	17 (45.95%)
Extremely helpful	9 (24.32%)
Not sure	2 (5.41%)

Moving on to why students utilized LLMs during the course, we asked them to indicate the specific purposes for which they used these tools. As shown in Table 11, the most frequently selected purpose was completing assignments and labs, reported by 26 students, which corresponds to 70.2% of the participants. This was followed by better understanding course concepts, selected by 22 students (59.4%). Exam preparation and understanding examples covered in class were each reported by 18 students, representing 48.6%. Additionally, 14 students (37.8%) used LLMs to review course content, while 12 students (32.4%) used them to generate examples related to course concepts. Use for VM setup tasks such as VirtualBox, VMware, Parallels, or FLARE VM was reported by 11 students (29.7%), and 10 students (27%) used LLMs for their final course project. Lastly, 9 students (24.3%) indicated that they used LLMs to discover new analysis techniques or methods. Interestingly, 7 of these 9 students were MSc students, suggesting that graduate students tend to use LLMs beyond course-related tasks and for exploratory purposes. Notably, all students reported using LLMs during the course, indicating broad and consistent adoption. To gain further insight into how LLMs were integrated into students' regular workflows, we asked whether they used these tools to get feedback on their assignment or lab answers. As shown in Table 12, the responses were fairly balanced, where 19 students (51.3%) reported using LLMs for this purpose, while 16 students (43.2%) indicated they did not. This finding is particularly important when compared with the broader use of LLMs for completing assignments and labs, which was the most commonly reported purpose overall, selected by 26 students (70.2%). The difference may suggest that 7 students (18.9%) used LLMs during assignments not to obtain direct feedback on their answers, but instead for more general support. These students may have relied on LLMs for guidance, clarification of assignment requirements, or assistance in specific questions related to the assignment, rather than seeking evaluation or validation of their full responses. Table 13 summarizes student responses regarding their frequency of LLM usage during the course. The most frequently selected option was "a few times a week", which indicated a regular and ongoing use. The next two popular answers were "once a week" and "only for homework, assignments, and labs". These responses showed that LLMs were generally not limited to specific, isolated periods of use, but were instead integrated into students' study routines throughout the course. The small number of students who reported using LLMs only during exam week or solely for the course project indicates that most relied on these tools consistently throughout the course, rather than for short-term needs. Looking at academic background, we compared students from computing fields (Computer Science, Cybersecurity, and IT) with those from other disciplines. Students from a non-computing background (11) were more likely to use LLMs on a weekly basis: 9 reported using them at least once a week, and only 2 limited their use to homework or the course project. This pattern suggests that non-computing students may have turned to LLMs more regularly to support them in handling the course's technical content. By contrast, students with a computing background showed a more varied pattern. 10 reported frequent use, daily or weekly, but many restricted their use to assignments or exams, and a smaller group said they rarely or never used LLMs.

Table 11. Purposes for LLM use during the course

Purpose	Number of Participants
For assignments and labs (homework)	26 (70.27%)
To better understand course concepts	22 (59.46%)
For exam preparation	18 (48.65%)
To better understand the examples covered in class	18 (48.65%)

To review course content	14 (37.84%)
To generate examples related to course concepts	12 (32.43%)
VM setup (e.g., VirtualBox, VMware, Parallels, FLARE VM)	11 (29.73%)
For the final course project	10 (27.03%)
To discover new analysis techniques and/or methods	9 (24.32%)
I did not use an LLM	0 (0.00%)

Table 12. Use of LLMs for getting feedback on assignments and/or labs

Student Response	Number of Participants
Yes	19 (51.35%)
No	16 (43.24%)
Not sure	2 (5.41%)

Table 13. Frequency of LLM use during the course

Student Response	Number of Participants
Daily	2 (5.41%)
A few times a week	14 (37.84%)
Once a week	8 (21.62%)
Rarely or never	3 (8.11%)
Only for homework, assignments, and labs	7 (18.92%)
Only during exam week	2 (5.41%)
Only for the course project	1 (2.70%)

B. *Effectiveness and Satisfaction*

To evaluate students' overall satisfaction with their experience using LLMs during the course, we asked them to rate their satisfaction on a 5-point scale, from "not satisfied at all" to "extremely satisfied". As illustrated in Table 14, the majority of students reported positive experiences. 25 students (67.5%) selected either "satisfied" or "extremely satisfied", suggesting that students were mostly pleased with how LLMs helped them in completing tasks and understanding course content. The absence of negative responses reinforces the notion that LLMs had a positive impact on the learning experience, even if the overall level of satisfaction varied among participating students.

After understanding the general satisfaction level among students, we specifically asked students how helpful LLMs were in terms of completing course-related tasks. As shown in Table 15, when asked about overall helpfulness in completing course-related tasks, most students reported a moderate to high level of helpfulness. The most common response was "somewhat helpful" with 17 students (45.9%), followed by "helpful" with 8 students (21.6%) and "extremely helpful" with 7 students (18.9%). The fact that nearly half of the students selected a neutral response may indicate that while LLMs were generally helpful, their effectiveness was not consistent across all tasks. This inconsistency could be attributed to students' limited use of prompt engineering techniques to clearly define and guide the tasks. We then asked students to narrow their evaluation to reverse engineering context, including advanced static and advanced dynamic analysis. As seen in Table 16, responses remained largely positive but varied more. A total of 20 students (54%) rated LLMs as either helpful or extremely helpful, while 17 students (45.9%) gave neutral or slightly positive ratings. Notably, no student rated LLMs as "not helpful at all," showing that LLMs provided at least some level of assistance in reverse engineering-related tasks. The overall satisfaction ratings, perceived usefulness in course-related and reverse engineering tasks, and the reported positive effects collectively indicate that students generally found LLMs beneficial, especially in simplifying and clarifying complex malware analysis tasks. Lastly, we asked students how likely they were to continue using LLMs for cybersecurity-related tasks in the future. As shown in Table 17, responses were significantly positive. A significant majority of students expressed a strong intention to continue using LLMs, with 48.6% (18 students) stating they are extremely likely to do so, and an additional 32.4% (12 students) indicating they are likely to incorporate LLMs into their future work. These results support the previous findings on helpfulness and satisfaction, suggesting that students not only found LLMs useful during the course but are also likely to use them in their future cybersecurity work.

Table 14. Satisfaction ratings of students using LLMs during the malware analysis course

Student Response	Number of Participants
Not satisfied at all	0 (0.00%)
Slightly satisfied	2 (5.41%)
Somewhat satisfied	10 (27.03%)
Satisfied	19 (51.35%)
Extremely satisfied	6 (16.22%)

Table 15. Student responses of how helpful LLMs were in terms of completing course-related tasks

Student Response	Number of Participants
Not helpful at all	2 (5.41%)
Slightly helpful	3 (8.11%)
Somewhat helpful	17 (45.95%)
Helpful	8 (21.62%)
Extremely helpful	7 (18.92%)

Table 16. Student responses of LLM helpfulness in answering reverse engineering-related questions (advanced static/dynamic analysis)

Student Response	Number of Participants
Not helpful at all	0 (0.00%)
Slightly helpful	6 (16.22%)
Somewhat helpful	11 (29.73%)
Helpful	12 (32.43%)
Extremely helpful	8 (21.62%)

Table 17. Student responses to how likely they are to continue using LLMs for cybersecurity-related tasks in the future

Student Response	Number of Participants
Not likely at all	2 (5.41%)
Slightly likely	1 (2.70%)
Somewhat likely	4 (10.81%)
Likely	12 (32.43%)
Extremely likely	18 (48.65%)

C. *Challenges and Issues*

To identify potential LLM limitations, we first asked students whether they encountered any issues while using LLMs during the malware analysis course. As shown in Table 18, the majority of students (59.4%) reported encountering no significant issues while using LLMs. However, 24.3% (9 students) selected "not sure", reflecting a level of uncertainty or mixed experiences, potentially deriving from minor challenges that were not substantial enough to be clearly recognized as problems. To follow up, we asked those who encountered issues to briefly describe the nature of the problems they faced. While the responses were varied and not easily grouped, a few important issues emerged. Some students mentioned that LLMs provided misinformation, gave overly generic answers to advanced

questions, or failed to offer sufficient help. Others reported token limitations and difficulties in processing large amounts of data. There were also instances where LLMs did not assist to students' questions because of ethical restrictions. Three students explicitly stated that they did not face any issues in this open-ended question. Lastly, we asked students whether they had observed refusals from LLMs to provide help due to ethical concerns, a known limitation when prompting LLMs with security-related questions. As shown in Table 19, 10 students (27%) reported experiencing such refusals, while 22 (59.4%) students said they had not, and 5 were unsure. The fact that nearly a third of the students did encounter such refusals highlights a potentially important limitation when relying on LLMs for malware analysis tasks. One potential solution is to improve prompt formulation by clearly stating its educational or research-oriented purpose, which may help LLMs respond more effectively to security-related queries. Another approach is to use locally hosted models, such as DeepSeek, which may bypass some of the restrictions and safeguards that limit malware analysis capabilities in commercial LLMs. Local models typically offer greater flexibility, as they are less constrained by safety filters designed to prevent misuse.

Table 18. Student responses to whether they encountered any issues while using LLMs during the malware analysis course

Student Response	Number of Participants
Yes	6 (16.22%)
No	22 (59.46%)
Not sure	9 (24.32%)

Table 19. Student responses on whether they experienced LLM refusals due to ethical concerns during the malware analysis course

Student Response	Number of Participants
Yes	10 (27.03%)
No	22 (59.46%)
Not sure	5 (13.51%)

D. Recommendations for improvement

To evaluate whether students found LLMs useful enough to recommend, we asked whether they would suggest using LLMs for the malware analysis course to other students or cybersecurity experts during their analysis. Many students responded positively, highlighting several key benefits. These included understanding concepts better, making reverse engineering and assembly code interpretation easier, and improving productivity. Some students also noted that LLMs helped them connect multiple points and recall key information more effectively. However, a few students were more cautious in their recommendations. They noted that LLMs can generate random or hallucinated responses and emphasized the importance of using these tools in a supervised manner. Some advised against overreliance, suggesting that LLMs should be treated as a supportive tool, not as the main method of analysis. Ethical and safety constraints were also highlighted as limitations, with one student specifically recommending the use of DeepSeek in cases where ChatGPT's restrictions hindered access to relevant information, citing its comparatively relaxed safeguards as an advantage in certain malware analysis scenarios. We also asked students what features or capabilities they would like to see in LLMs to make malware analysis easier. One of the most frequently mentioned suggestions was integrating LLMs into existing malware analysis tools, particularly IDA Pro. For example, one student expressed interest in a setup similar to the Google Colab–Gemini integration, where the LLM could provide real-time assistance. Others suggested using LLMs to automate code deobfuscation, disassembly interpretation, and binary behavior analysis, which would reduce the manual effort. Additionally, another suggested feature was having LLMs generate brief summaries of malware behavior, highlighting suspicious files, URLs, or key indicators. Some students also wanted LLMs to be able to run malware in sandbox environments to support dynamic analysis.

E. Discussion

Our findings reveal that every student was familiar with LLM tools and used at least one during the course, demonstrating widespread awareness and adoption of these technologies in the context of malware analysis education. The fact that LLMs are used and found helpful even in this niche, highly specialized context, like malware analysis, suggest that these tools have the potential to be useful in complex cybersecurity workflows, not just in general academic tasks. In our case, students perceived LLMs as useful for completing demanding tasks such as static analysis and reverse engineering. This pattern reflects the Technology Acceptance Model (TAM), which emphasizes that adoption is largely driven by perceived usefulness and ease of use [22]. To improve the usefulness of LLM tools in the malware analysis domain, a key step would be integrating LLMs into already established malware analysis tools. Based on our results, IDA Pro and OllyDbg were the two tools whose outputs students most frequently analyzed with LLMs, and the majority expressed a desire to see direct LLM integration with these platforms. Also importantly, advanced static analysis was one of the two analysis types where students reported using LLMs the most. Collectively, these findings indicate that interpreting disassembled code and performing reverse engineering are among the most challenging aspects of the course for students. This likely explains the frequent use of LLMs to help interpret and clarify outputs from tools like IDA Pro and OllyDbg. By integrating LLM into these tools, we can diminish the complexity of reverse engineering and help students better understand disassembled code without relying solely on manual interpretation. Building on this LLM integration concept, a possible direction could be the development of dedicated plug-ins by cybersecurity experts that integrate LLMs into malware analysis tools in a structured and user-friendly way. Based on our results, the majority of the students who used LLMs during the malware analysis course were already satisfied with their experiences, and found LLMs helpful in terms of completing course-related tasks and answering reverse engineering-related questions. However, students also faced several challenges while using LLMs during the malware analysis course. These issues include token limitations, difficulties in processing large inputs, hallucinated responses, and ethical safeguard restrictions. In addition to these challenges, students often need to write lengthy prompts to explain their questions clearly to the LLM. These issues highlight an opportunity for improvement through well-designed tool plug-ins. Plug-ins developed by

cybersecurity experts could minimize the need for manual prompting, provide context-specific assistance directly during analysis and solve some of the limitations discussed, such as token size limits and ethical safeguard restrictions. By integrating LLM functionality into the tools students already use, like IDA Pro and OllyDbg, these plug-ins could offer more accurate, task-specific support, improving both the learning experience and the malware analysis process. Beyond LLM integration into tools and plug-in development, another direction that can be explored is malware-specific LLMs. Popular, general-purpose LLMs can be helpful in many cases, but they often fail to deliver accurate or correct information in complex malware analysis tasks, or give generic responses to advanced questions. Similar limitations have been observed in software engineering, where ChatGPT performs well on many tasks but fails to provide accurate answers for specialized activities such as code vulnerability detection and information retrieval-based test prioritization [23]. A fine-tuned LLM trained on real malware samples and their analysis could provide more reliable, domain-specific explanations to questions, and catch more details than a general-purpose LLM. In addition, a domain-specific model could streamline the malware analysis process by delivering targeted support based on the specific type of analysis being performed. By providing clear, context-aware explanations and incorporating both course material and real-world examples, such a model would enhance the learning experience and make complex tasks more approachable for students. Furthermore, its specialized focus would help reduce noise, resulting in clearer, more accurate and more actionable insights. According to our survey results, the second and third most selected purposes for using LLMs were to better understand course topics and to prepare for the exam. This indicates that students used these tools not only for solving specific analysis tasks, but also for grasping theoretical content and reinforcing their learning, which were reasons commonly mentioned when recommending the use of LLMs for the malware analysis course. Also, over half of the students participating in our survey reported that they have used LLMs to get feedback on assignments and labs. These findings highlight the potential for custom course-specific LLM tutors that support students to better understand malware analysis course content. Such a tool would allow students to revisit lecture content in a low-pressure environment, ask clarifying questions, and receive immediate feedback tailored to their individual needs. This also aligns with previous studies emphasizing the usefulness of LLMs in offering feedback and personalized resources [12–14]. While such technical solutions hold promise, their effectiveness ultimately depends on how responsibly LLMs are embedded into teaching and learning practices in malware analysis education. When answering the open-ended question of whether students would recommend using LLMs in their future course-related tasks, some raised concerns about its overreliance and unsupervised use. Some recent discussions in higher education suggest returning to pen-and-paper examinations or designing tasks solely to block LLM use because of academic integrity [24]. However, according to our survey, all students have engaged with LLMs at some point in the course, and at least half used them to get feedback on assignments. This level of adoption suggests that prohibiting or discouraging LLM use is not feasible. Instead, alternative approaches should be considered, such as redesigning assessments into formats less easily replicated by LLMs, including group work or oral presentations [24]. Another step could be to encourage students to be transparent and to disclose where and how they used LLMs to prevent academic dishonesty.

Finally, lecturers can play a key role by teaching students how to use LLMs more effectively, for instance, by introducing successful prompt engineering techniques and task-specific querying strategies to achieve malware analysis-related tasks. Importantly, LLMs should be framed as support tools rather than substitutes for students' own analysis. This approach reflects constructivist learning theory, which emphasizes that knowledge is built through active engagement with tasks rather than passive reception. In this sense, educators who integrate AI tools like ChatGPT into classrooms can help boost student engagement and support deeper learning [25]. Framing LLMs as support tools rather than substitutes also helps prevent overreliance and ensures that students continue to develop essential skills in reverse engineering and critical evaluation. As recent work highlights, effective use of LLMs depends heavily on prompt quality, making it a tool that requires expertise to be applied successfully [26]. LLMs are already successful in many malware analysis tasks, such as reverse engineering and malware deobfuscation. For example, recent studies have shown that ChatGPT is capable of detecting and deobfuscating request bodies [1], reinforcing its value in static analysis tasks. This is particularly encouraging given our own findings, which show that students most frequently turned to LLMs for support in static analysis tasks, where such capabilities are particularly valuable. By proactively incorporating LLM usage into course design, demonstrating these real-world use cases, and guiding students to develop better prompt engineering practices, educators can help students get even more out of these tools, boosting both overall satisfaction and the practical effectiveness of students' learning experience.

V. Conclusion

This study surveyed 37 students to examine how they use LLMs during their malware analysis course, focusing on their purposes of use, frequency, perceived helpfulness, and encountered challenges. The results show that all students used LLMs throughout the course, with ChatGPT as the dominant choice. The most popular reasons students use LLMs were getting help for assignments and labs (70.2%), and to better understand course concepts (59.4%). Students most frequently analyzed the output of the malware analysis tool IDA Pro (83.7%) with LLMs, followed by OllyDbg and Wireshark (43.2%). The majority of students (81%) expressed their intention to continue using LLMs for cybersecurity-related tasks in the future. While 59.4% of students reported no significant issues, 27% experienced refusals to answer due to ethical safeguards. Other challanges included misinformation, overly generic responses, token/input size limitations, and difficulty handling complex prompts. Students also shared practical recommendations for improvement. These included tighter integration with tools such as IDA Pro, automation of disassembly interpretation

and deobfuscation, generation of behavioral summaries, and dynamic analysis support. 78.3% expressed interest in LLM integration with IDA Pro, and 35.1% with OllyDbg, further reinforcing the need for context-aware tool augmentation. To summarize, our results showed that students found LLMs helpful and valuable across both technical and learning tasks in the malware analysis course. Their effectiveness could be further improved through better LLM integration into existing tools, expert-developed plug-ins, malware-specific models, LLM tutors, and lecturers' guidance on effective prompting techniques to enhance learning outcomes. Going forward, exploring redesigned assessment formats and encouraging transparency in LLM use could support academic integrity and responsible use.

It is important to recognize the limitations of this study. First, the sample size was small (37 students). This may limit how far the results can be generalized to broader student populations. Second, the survey was conducted at a single institution, Sabancı University. The findings may not capture the experiences of students in malware analysis courses at other universities. Third, the data were collected from a single survey, which may introduce bias or lead to misinterpretation of the questions. Finally, the study focused on a limited set of tools and models available during the course. Future advances in LLMs could shape students' perceptions and experiences differently.

Future work could expand this study in several ways. Conducting multi-institutional surveys with larger student groups taking malware analysis courses would strengthen the validity and applicability of the findings. Moreover, similar investigations could be extended to other technical cybersecurity courses, such as penetration testing, to explore how LLMs support different tools and specialized domains. Controlled experiments could be designed to compare course outcomes between students who use LLMs and those who do not, offering stronger evidence of their impact on learning and performance. Real tool integration, such as plug-ins for IDA Pro, OllyDbg, or WireShark, could also be developed and tested to evaluate their usefulness and efficiency in supporting students' malware analysis workflows.

APPENDIX

A. Survey Questions

1) Section 1: Demographics

A1. What is your gender?

A2. What is your age?

A3. Are you currently working or doing an internship related to cyber security?

A4. What is your major or area of study?

A5. What is your current grade level or academic standing?

A6. Apart from your malware analysis course, how many cybersecurity courses have you taken?

2) Section 2: LLM usage

A7. Did you utilize an LLM during the malware analysis course? If yes, please specify which model(s) you used.

A8. In which type of analysis did you use an LLM the most?

A9. In which type of analysis did you use an LLM the least?

A10. Which tool's output did you analyze using an LLM?

A11. For which tool would you like to integrate an LLM to better understand its output?

A12. What was your purpose for using an LLM during the course?

A13. How helpful was the LLM in understanding or explaining the behavior of disassembled code (e.g., Assembly code displayed in IDA Pro)?

A14. Did you use an LLM to get feedback on your assignment and/or lab answers?

A15. How often did you use an LLM during the malware analysis course?

A16. Did the LLM help you complete your reverse engineering analysis more efficiently or effectively?

A17. On a scale of 1-5, how helpful was the LLM in assisting reverse engineering (advanced static/dynamic analysis) related questions?

A18. How helpful was the LLM in terms of completing your tasks?

A19. Have you observed refusal to help due to ethical concerns when prompting LLMs for malware analysis course?

A20. Based on your experience, would you recommend using the LLM for malware analysis course to other students and cybersecurity experts during their analysis? Why?

A21. How likely are you to continue using LLMs for cybersecurity-related tasks in the future?

A22. On a scale of 1–5, how satisfied were you with your experience using an LLM for completing tasks and understanding topics related to the course?

A23. Did you encounter any issues while using the LLM during the course?

A24. If you encountered any issues while using an LLM in the malware analysis course, please describe them briefly.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

OC conceptualized the study, conducted the research, curated and analyzed the data, supervised the overall research process, and edited the paper. NB prepared the data tables, contributed to the methodological design, and wrote the paper; all authors had approved the final version.

REFERENCES

[1] M. B. Ozkok, B. Birinci, O. Cetin, B. Arief, and J. Hernandez-Castro, "Honeypot's best friend? Investigating ChatGPT's ability to evaluate honeypot logs," in *Proc. the 2024 European Interdisciplinary Cybersecurity Conference*, 2024, pp. 128–135.

[2] G. Desolda, F. Greco, and L. Viganò, "APOLLO: A GPT-based tool to detect phishing emails and generate explanations that warn users," in *Proc. the ACM on Human-Computer Interaction*, vol. 9, no. 4, pp. 1–33, 2025.

[3] A. L. Martínez, A. Cano, and A. Ruiz-Martínez, "Generative artificial intelligence-supported pentesting: A comparison between claude opus, gpt-4, and copilot," arXiv preprint arXiv:2501.06963, *2025.*

[4] H. Jelodar, S. Bai, P. Hamedi, H. Mohammadian, R. Razavi-Far, and A. Ghorbani, "Large Language Model (LLM) for software security: Code analysis, malware analysis, reverse engineering," arXiv preprint arXiv:2504.07137, 2025.

[5] O. Çetin, E. Ekmekcioglu, B. Arief, and J. Hernandez-Castro, "An empirical evaluation of large language models in static code analysis

for PHP vulnerability detection," *Journal of Universal Computer Science*, vol. 30, no. 9, 2024.

[6] O. Çetin, B. Birinci, Ç. Uysal, and B. Arief, "Exploring the cybercrime potential of llms: A focus on phishing and malware generation," in *Proc. European Interdisciplinary Cybersecurity Conference*, Springer, 2025.

[7] M. Amoozadeh, D. Daniels, D. Nam, A. Kumar, S. Chen, M. Hilton, S. Srinivasa Ragavan, and M. A. Alipour, "Trust in Generative AI among students: An exploratory study," *in Proc. the 55th ACM Technical Symposium on Computer Science Education V. 1*, 2024.

[8] R. Wu and Z. Yu, "Do AI chatbots improve students learning outcomes? Evidence from a meta-analysis," *British Journal of Educational Technology,* Wiley Online Library, vol. 55, no. 1, pp. 10–33, 2024.

[9] L. Wang and W. Li, "The impact of AI usage on university students' willingness for autonomous learning," *Behavioral Sciences*, vol. 14, no. 10, p. 956, 2024.

[10] O. Boubker, "From chatting to self-educating: Can AI tools boost student learning outcomes?" *Expert Systems with Applications*, vol. 238, 121820, 2024.

[11] K. Hanifi, O. Cetin, and C. Yilmaz, "On chatgpt: Perspectives from software engineering students," presented at 2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security (QRS), IEEE, 2023.

[12] H. B. Essel, D. Vlachopoulos, A. Tachie-Menson, E. E. Johnson, and P. K. Baah, "The impact of a virtual teaching assistant (chatbot) on students' learning in Ghanaian higher education," *International Journal of Educational Technology in Higher Education*, vol. 19, no. 1, p. 57, 2022.

[13] Y. Chen, S. Jensen, L. J. Albert, S. Gupta, and T. Lee, "Artificial Intelligence (AI) student assistants in the classroom: Designing chatbots to support student success," *Information Systems Frontiers*, vol. 25, no. 1, 2023.

[14] A. Viorennita, L. Dewi, and C. Riyana, "The role of ChatGPT AI in student learning experience," *Indonesian Journal of Multidiciplinary Research*, vol. 3, no. 2, pp. 445–452, 2023.

[15] M. Maulana, C. Darmawan, and R. Rahmat, "The use of ChatGPT in educational review from the perspective of academic ethics," *Bhineka Tunggal Ika: Journal of Theory and Practice of Civic Education*, vol. 10, pp. 58–66, May 2023.

[16] L. J. Quintans-J´unior, R. Q. Gurgel, A. A. d. S. Ara´ujo, D. Correia, and P. R. Martins-Filho, "ChatGPT: The new panacea of the academic world," Revista da Sociedade Brasileira de Medicina Tropical, vol. 56, 2023.

[17] A. Guilherme, "AI and education: the importance of teacher and student relations," *AI & Society*, vol. 34, no. 1, 2019.

[18] N. D. Nguyen, "Exploring the role of AI in education," *London Journal of Social Sciences*, no. 6, pp. 84–95, 2023.

[19] M. R. Hawa, M. Owda, and A. Y. Owda, "Enhancing digital investigation: The role of generative AI (ChatGPT) in evidence identification and analysis in digital forensics," in *Proc. 2025 12th International Conference on Information Technology (ICIT)*, IEEE, 2025, pp. 19–26.

[20] T. Espinha Gasiba, A.-C. Iosif, I. Kessba, S. Amburi, U. Lechner, and M. Pinto-Albuquerque, "May the source be with you: On chatgpt, cybersecurity, and secure coding," *Information*, vol. 15, no. 9, p. 572, 2024.

[21] M. Fu, J. Pasuksmit, and C. Tantithamthavorn, "AI for devsecops: A landscape and future opportunities," *ACM Transactions on Software Engineering and Methodology*, vol. 34, no. 4, pp. 1–61, 2025.

[22] K. Li, "Determinants of college students' actual use of AI-based systems: An extension of the technology acceptance model," *Sustainability*, vol. 15, no. 6, p. 5221, 2023.

[23] G. Sridhara, S. Mazumdar, and others, "Chatgpt: A study on its utility for ubiquitous software engineering tasks," arXiv preprint arXiv:2305.16837, 2023.

[24] M. Sullivan, A. Kelly, and P. McLaughlan, "Chatgpt in higher education: Considerations for academic integrity and student learning," *Journal of Applied Learning & Teaching*, vol. 6, no. 1, pp. 31–40, 2023.

[25] S. Grubaugh, G. Levitt, and D. Deever, "Harnessing AI to power constructivist learning: An evolution in educational methodologies," *EIKI Journal of Effective Teaching Methods*, vol. 1, no. 3, 2023.

[26] L. J. Jacobsen and K. E. Weber, *The Promises and Pitfalls of ChatGPT as a Feedback Provider in Higher Education: An Exploratory Study of Prompt Engineering and the Quality of AI-Driven Feedback*, OSF preprints, 2023.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).