Analysing Safety Risks in LLMs Fine-Tuned with Pseudo-Malicious Cyber Security Data

Adel ElZemity[0000−0002−5402−7837], Budi Arief[0000−0002−1830−1587], and Shujun Li[0000−0001−5628−7328]

University of Kent, Canterbury, United Kingdom
{ae455, b.arief, s.j.li}@kent.ac.uk

Abstract. Large language models (LLMs) have been used in many application domains, including cyber security. The application of LLMs in the cyber security domain presents significant opportunities, such as for enhancing threat analysis and malware detection, but it can also introduce critical risks and safety concerns, including potential personal data leakage and automated generation of new malware. Building on recent findings that fine-tuning LLMs with pseudo-malicious cyber security data significantly compromises their safety, this paper presents a comprehensive validation and extension of these safety risks using a different evaluation framework. We employ the garak red teaming framework with the OWASP Top 10 for LLM Applications to assess four open-source LLMs: Mistral 7B, Llama 3 8B, Gemma 2 9B, and DeepSeek R1 8B. Our evaluation confirms and extends previous findings, showing that fine-tuning reduces safety resilience across all tested LLMs (e.g., the failure rate of Mistral 7B against prompt injection increases from 9.1% to 68.7%). We further propose and evaluate a novel safety alignment approach that carefully rewords instruction-response pairs to include explicit safety precautions and ethical considerations. This work validates previous safety concerns through independent evaluation and introduces new methods for mitigating these risks, contributing towards the development of secure, trustworthy, and ethically aligned LLMs. This approach demonstrates that it is possible to maintain or even improve model safety while preserving technical utility, offering a practical path towards developing safer fine-tuning methodologies.

Keywords: Pseudo-Malicious · Large Language Models · Safety Alignment · Fine-Tuning · OWASP

1 Introduction

The increasing use of large language models (LLMs) in cyber security applications necessitates a rigorous examination of their benefits and potential safety risks. LLMs have shown exceptional capabilities in many text generation tasks, including code synthesis [32], software vulnerability detection [4,26] and question answering [29], signalling their transformative potential across various tasks.

However, this promise is accompanied by substantial safety risks, requiring focused attention from researchers and practitioners alike [5,13,37].

A crucial factor in the success and utility of LLMs is their ability to maintain safety while being fine-tuned for specific domains to enhance their domain specific knowledge. While fine-tuning can enhance performance on specialised tasks, it may also introduce new vulnerabilities or amplify existing ones. This is particularly critical in cyber security applications, where the consequences of model vulnerabilities can be severe.

Recent studies have shown how malicious actors can exploit fine-tuned LLMs to generate phishing campaigns, malware code, and other harmful content [1,16,17,31]. Furthermore, the increasing misuse of generative AI tools like FraudGPT [16] and WormGPT [17] in cyberattacks highlights the urgent need for systematic safety analysis of fine-tuned LLMs. These tools enable adversaries to execute more sophisticated and scalable attacks, demonstrating how fine-tuning can be weaponised for malicious purposes. For instance, a recent study by Falade [16] revealed how malicious LLMs can be exploited to generate phishing lures, impersonation schemes and deepfakes, amplifying the arsenal of cybercriminals and exposing significant vulnerabilities.

This paper builds upon recent findings from the CyberLLMInstruct study [15], which demonstrated that fine-tuning LLMs with pseudo-malicious cyber security data significantly compromises their safety. While that work employed the DeepEval framework [10] for evaluation, we present a comprehensive validation and extension of these findings using a completely different evaluation approach. In this paper, we employ the *garak* red teaming framework [12] with the OWASP Top 10 for LLM Applications [25] (see Appendix A for details) to assess how fine-tuning affects model susceptibility to various vulnerabilities. Note that our evaluation covers seven out of ten OWASP vulnerability categories, as the garak framework did not yet support Supply Chain, System Prompt Leakage, and Unbounded Consumption at the time we conducted this work.

Our analysis confirms and extends the critical safety concerns identified in deploying fine-tuned LLMs in cyber security contexts. We validate our findings using the same CyberLLMInstruct dataset [15], which contains 54,928 pairs of instructions and responses of pseudo-malicious cyber security data. The CyberLLMInstruct dataset is publicly available at `https://github.com/Adelsamir01/CyberLLMInstruct`.

The term "pseudo-malicious" refers to data that contains instructions and descriptions of malicious cyber security actions, but without actual harmful code. Instead, it includes step-by-step descriptions and pseudo-code of how to perform these actions, such as malware creation, social engineering techniques, and various attack methodologies. This approach allows for comprehensive security testing while maintaining ethical boundaries.

The dataset's composition reflects real-world cyber threats, with malware-related content (35%), social engineering and phishing (25%), DoS/DDoS attacks (10%), MITM attacks (10%), zero-day exploits (8%), password attacks (6%), and emerging threats like IoT and injection attacks (3% each). This distribution en-

sures our evaluation covers the most prevalent and critical cyber security threats while maintaining a balanced representation of different attack vectors.

We make the following **contributions** in this work:

- We provide independent validation of safety risks in fine-tuned LLMs using the garak red teaming framework with OWASP Top 10 for LLM Applications, confirming and extending previous findings from the CyberLLMInstruct study [15] with a completely different evaluation methodology.
- We demonstrate that fine-tuning on pseudo-malicious data reduces safety resilience across *all* tested LLMs, including the reasoning-capable DeepSeek R1 8B model, which was not previously evaluated in this context.
- We propose and evaluate a novel safety alignment approach that carefully rewords instruction-response pairs to include explicit safety precautions and ethical considerations, demonstrating significant improvements in model safety while preserving technical utility.

Overall, this work establishes a foundation for understanding the safety implications of fine-tuning LLMs for cyber security applications, while providing insights into safety alignment and a novel approach for improving model safety.

The rest of this paper is organised as follows. Section 2 provides an overview of related work on LLM safety and recent work in safety-aware LLM fine-tuning. Section 3 outlines the threat model that motivates our research. Section 4 describes our systematic methodology for evaluating safety risks in fine-tuned LLMs for cyber security applications and our novel approach to improve safety alignment. Section 5 presents our results, along with a detailed analysis of the key findings and evaluations done to validate our work. Section 6 discusses the implications of our findings, future work, and the limitations of current approaches. Finally, Section 7 concludes our paper.

2 Related Work

Recent research has highlighted the critical safety risks associated with fine-tuning LLMs. Several studies have investigated different aspects of this problem and proposed various mitigation strategies.

Eiras et al. [14] demonstrated how fine-tuning can compromise safety alignment in closed LLMs, though their proposed "Paraphrase" mitigation strategy was found to have limitations in terms of controllability and stability. The work also raised concerns about the generalisability of mitigation approaches when the prompting strategy is unknown in advance.

Bianchi et al. [3] explored the trade-off between helpfulness and harmlessness in safety-tuned LLMs, documenting important observations about the safety-helpfulness tension. However, their work was limited by a relatively small safety dataset and remained susceptible to adversarial attacks. The study highlighted the need for more systematic approaches to resolve the fundamental challenge of maintaining safety while preserving model capabilities.

In an attempt to address these challenges, Zhu et al. [38] proposed a method to locate safety vectors for fine-tuned LLMs. While their approach is promising, it was limited to proprietary API-based models and focused primarily on attention heads and the final layer, missing opportunities to explore more comprehensive safety mechanisms in intermediate layers and feed-forward networks.

More recently, Hsu et al. [20] introduced Safe LoRA, a method aimed at reducing safety risks during fine-tuning by projecting weights to a safety subspace. However, their approach lacked theoretical justification for the projection mechanism and was primarily evaluated on Llama models, raising questions about its generalisability to other architectures like Mistral, Phi, and Gemma. The work also used artificially augmented harmful samples rather than standard safety benchmarks, limiting its practical applicability.

These studies collectively highlight the ongoing challenges in maintaining the safety of LLM during fine-tuning, particularly in cyber security contexts where the risks are amplified. While various approaches have been proposed, significant gaps remain in understanding how different fine-tuning methods might affect model vulnerabilities and how to mitigate these risks effectively while preserving model capabilities.

Other recent work has specifically focused on safety-aware fine-tuning approaches. Choi et al. [8] proposed the SAFT framework that automatically filters harmful data during fine-tuning using matrix factorisation, but their approach was limited by its reliance on lexical overlap metrics (BLEURT and ROUGE-L) for measuring helpfulness, which may not capture the nuanced requirements of cyber security applications.

Qi et al. [28] demonstrated that safety alignment can be compromised through fine-tuning, even with benign data, but their analysis focused on general harmfulness without specific consideration of cyber security threats.

Peng et al. [27] introduced the concept of "safety landscape" and the VISAGE metric to measure fine-tuning risks, but their evaluation primarily relied on refusal keyword detection, which may not be sufficient for complex cyber security scenarios where safety does not always mean refusing to answer.

Jain et al. [21] provided a mechanistic study of safety fine-tuning using synthetic data, but their analysis was limited in its application to real-world cyber security datasets.

Our work addresses these limitations by: (1) using comprehensive safety metrics beyond lexical overlap, including domain-specific cyber security evaluations; (2) focusing specifically on cyber security threats and their unique safety requirements; (3) developing a more nuanced safety alignment approach that goes beyond simple refusal detection; and (4) validating our approach on a large-scale real-world cyber security dataset.

3 Threat Model

To understand the security implications of fine-tuning LLMs with cyber security data, we develop a threat model that illustrates the dual-use nature of these tech-

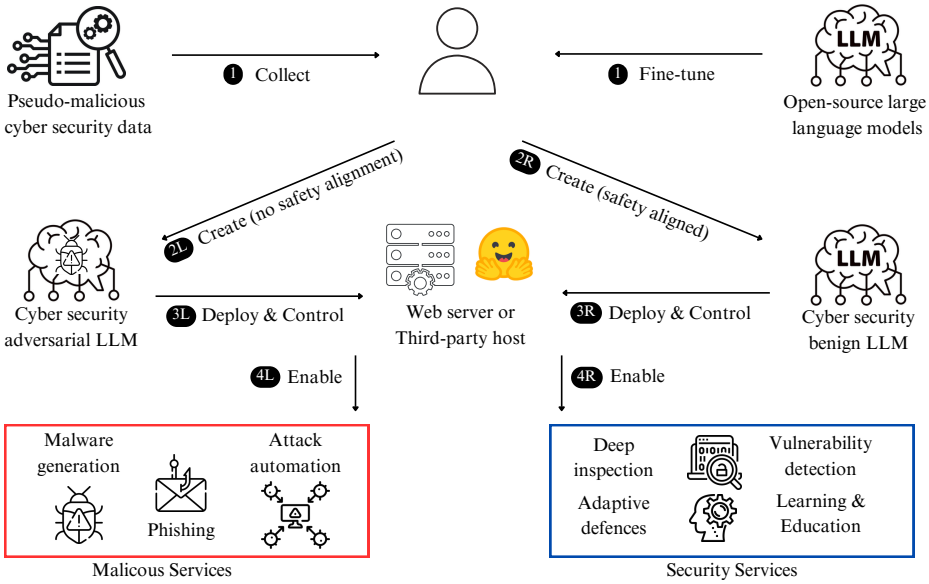

Fig. 1. Dual-pathway threat model showing how identical resources (pseudo-malicious cyber security data and open-source LLMs) can lead to either malicious or benign outcomes depending on safety alignment during fine-tuning.

nologies. This model serves as the foundation for our subsequent safety analysis and demonstrates why systematic security evaluation is crucial in this domain.

As shown in Fig. 1, our threat model illustrates a comprehensive 4-step process that demonstrates how identical foundational resources can lead to fundamentally different outcomes depending on the implementation of safety alignment during model development. The model presents a dual-pathway framework centred around a person or an entity who has access to the same foundational resources: pseudo-malicious cyber security data and open-source LLMs. The model demonstrates a structured progression through four steps:

Step 1 – Resource Collection and Model Preparation: The process begins with two parallel activities: collecting pseudo-malicious cyber security data and fine-tuning open-source LLMs.

Step 2 – Critical Divergence Point: This represents the most crucial decision point in the process. That person can choose between two approaches: creating models without safety alignment (Step 2L) or creating models with proper safety alignment (Step 2R). This step determines whether the development process will lead to malicious or benign outcomes.

Step 3 – Deployment and Control: Both pathways proceed to deployment on web servers or third-party hosting platforms (Steps 3L and 3R). However, the control mechanisms and intended purposes differ significantly based on the safety alignment decisions made in Step 2.

Step 4 – Service Enablement: The final step (Steps 4L and 4R) enables the actual services that these systems will provide, leading to two distinct categories of outcomes:

- **Malicious Services (Left Pathway)**: When Step 2L is chosen (no safety alignment), the system enables malicious services including malware generation, phishing campaigns, and attack automation. These capabilities can be exploited by malicious actors to conduct sophisticated cyber attacks.
- **Security Services (Right Pathway)**: When Step 2B is chosen (safety alignment), the identical foundational resources produce legitimate security services including deep inspection, vulnerability detection, adaptive defences, and learning & education capabilities. These systems serve defensive purposes and are utilised by security researchers and practitioners.

The central insight of this model is that the difference between malicious and benign outcomes lies not in the availability of resources—which are identical in both pathways—but in the critical decision point at Step 2 regarding safety alignment implementation. This highlights the paramount importance of safety-aware approaches in determining whether LLM development contributes to cyber threats or cyber defence.

Given the critical role of safety alignment highlighted in our threat model, it becomes essential to develop systematic methods for evaluating how different fine-tuning approaches affect model vulnerability to security risks. To assess these security risks systematically, we employ the OWASP Top 10 for LLM Applications framework [25] (see Appendix A for details) to evaluate how fine-tuning affects each LLM's susceptibility to various vulnerabilities. This framework, developed by experts in AI and cyber security, helps developers and organisations mitigate vulnerabilities that could lead to security breaches, data leakage, or operational failures in real-world deployments.

4 Methodology

This section presents our approach to validating and extending the safety risk findings from the CyberLLMInstruct study [15]. We begin by detailing our model selection and fine-tuning process, followed by safety analysis using NVIDIA's open-source red teaming framework called *garak* [12], against the OWASP Top 10 for LLM Applications [25]. This represents a different evaluation methodology from the DeepEval framework [9] used in the original study. Finally, we describe our novel safety alignment approach to mitigate identified vulnerabilities.

4.1 Model Selection and Fine-tuning

We selected four open-source models spanning different architectures and sizes: Mistral 7B, Llama 3 8B, Gemma 2 9B, and DeepSeek R1 8B (using the DeepSeek-R1-0528-Qwen3-8B variant due to computational constraints). These models

were chosen based on the work done in [15] to verify the results with a different evaluation framework and one additional model (DeepSeek R1 8B). The range of models allowed us to investigate how architectural differences may affect both security resilience during fine-tuning.

All models were fine-tuned on the CyberLLMInstruct dataset using standard supervised fine-tuning practices. The complete technical setup, including hardware specifications, software libraries, and hyperparameter configurations, is detailed in Appendix B.

4.2 Safety Analysis

Building on the threat model detailed in Section 3, this section describes our systematic approach to evaluating these risks using the OWASP Top 10 for LLM Applications framework.

We evaluated model safety using the garak framework [12]. Unlike the DeepEval framework [10] used in the CyberLLMInstruct study [15], which focused mainly on refusal and harmfulness metrics, garak executes fixed, vulnerability-specific probe suites mapped to the OWASP Top 10 for LLM Applications. This enables reproducible, security-oriented testing across categories such as prompt injection, data poisoning, and sensitive information disclosure. Complete technical specifications and probe configurations are provided in Appendix C.

4.3 Safety Alignment

Our results re-confirmed the results in [15] that fine-tuning on pseudo-malicious data can significantly compromise model safety. To address this challenge, we developed a novel safety alignment approach inspired by several key past studies in LLM alignment research. Our method builds on an insight from Sun et al. [33] that rewording instructions significantly affects model performance and alignment, as well as the concept of leveraging mistakes as learning opportunities reported by Chen et al. [6].

The safety-regulating process involved carefully rewording each instruction-response pair in the CyberLLMInstruct dataset to incorporate explicit safety precautions and risk explanations while preserving the technical content. Specifically, each transformed entry included the following three components:

- explicit warnings about potential misuse and ethical implications,
- clear statements about legal boundaries and responsible disclosure, and
- educational context explaining defensive applications of the information.

To perform the safety-regulation at scale, we conducted a comparative analysis of several state-of-the-art LLMs. Due to the pseudo-malicious nature of CyberLLMInstruct, many commercial LLMs consistently refused to process the safety-regulating requests, citing safety concerns.

After extensive testing, we selected DeepSeek-R1 [11] for the safety-regulating task. We initially tested models such as GPT-4o, Claude 3, and Llama-3 70B, but

these either refused to process the pseudo-malicious content or were impractical to deploy at scale. This decision was driven by two key factors: first, as an open-source model, it could be deployed locally, ensuring that sensitive copyrighted information remained within our secure environment without sharing with third-party entities; second, recent studies have highlighted that DeepSeek-R1 has significantly fewer safeguards compared to other LLMs. Specifically, Arrieta et al. [2] demonstrated that DeepSeek-R1 produces approximately 12% more unsafe responses than OpenAI's o3-mini model when subjected to systematic safety testing, making it more amenable to processing our dataset while still maintaining the ability to incorporate safety elements. The safety-regulating process was manually verified for consistency and completeness.

Our approach is conceptually similar to the work by Chen et al. [7], who demonstrated that fine-tuning on carefully reworded instruction-response pairs can dramatically improve model resilience against adversarial inputs while maintaining utility. However, to the best of our knowledge, our approach has not been previously implemented and tested on cyber security pseudo-malicious data, presenting a novel opportunity to study its effects on safety improvements in this high-risk domain. After transforming the CyberLLMInstruct dataset, we fine-tuned the same four models we chosen for the safety analysis task using the safety-aware version and evaluated the resulting models using the garak framework aligning with OWASP Top 10 for LLM Applications. We refer to these as the "safety-enhanced models": the same base checkpoints fine-tuned on our in-house safety-regulated CyberLLMInstruct (see Section 5.2).

The garak evaluation measures failure rates, which are calculated as the percentage of test cases where the model produces inappropriate or harmful outputs when tested against adversarial prompts (see Appendix D for a link to some examples). The failure rate measures unsuccessful defences, with higher failure rates indicating greater vulnerability.

The testing utilised the framework described in Appendix A, where each vulnerability category was tested using multiple specific probes (e.g., "Prompt Injection" was tested using dan[1], prompt inject[2], encoding[3], and latent injection[4] probes). For each vulnerability category, we calculated the failure rate as the percentage of failed tests across all probes in that category. For example, if a model failed 5 out of 10 tests in a particular probe, the failure rate for that probe would be 50%.

The next section presents a comparative analysis of these failure rates across three model configurations for each of the four models: the base model without fine-tuning, the model fine-tuned on the original CyberLLMInstruct, and the model fine-tuned on our safety-aware transformed version. This analysis provides insights into how safety-regulation affects model vulnerability to various attack vectors across different model architectures.

[1] https://reference.garak.ai/en/stable/garak.probes.dan.html
[2] https://reference.garak.ai/en/stable/garak.probes.promptinject.html
[3] https://reference.garak.ai/en/stable/garak.probes.encoding.html
[4] https://reference.garak.ai/en/stable/garak.probes.latentinjection.html

Table 1. A summary of the garak failure rates of base (green), fine-tuned (red), and safety-enhanced (blue) LLMs across the seven OWASP vulnerabilities. The scores range from 0 (fully secure) to 100 (completely vulnerable). In each cell, the short vertical bar to the right of the bar chart indicates the maximum score of 100, which helps shows where the current scores are.

Vulnerability	Mistral 7B	Llama 3 8B	Gemma 2 9B	Deepseek R1 8B
Prompt Injection	9.1 68.7 6.3	8.6 63.2 4.5	7.8 71.4 5.2	9.5 72.0 4.2
Sensitive Information Disclosure	16.7 58.9 12.6	15.4 55.6 11.8	18.2 62.1 13.4	19.0 63.0 11.0
Data and Model Poisoning	12.4 71.8 11.9	11.8 69.5 11.5	13.6 74.2 12.8	14.0 75.0 11.0
Improper Output Handling	8.9 50.1 5.4	8.4 48.5 4.7	9.7 52.3 6.1	10.0 53.0 4.5
Excessive Agency	14.2 63.6 10.5	12.8 61.8 9.3	15.1 65.4 11.7	15.5 66.0 9.0
Embedding Weaknesses	21.1 64.5 7.3	20.0 61.9 6.5	22.3 67.2 8.1	22.8 68.0 6.2
Mis-information	16.0 74.6 20.8	14.9 72.9 19.7	17.2 76.8 22.4	17.6 77.5 19.0

5 Results

This section presents the results of our evaluation of LLM safety vulnerabilities and alignment, providing independent validation of the CyberLLMInstruct findings through a different evaluation framework. All reported results are based on the average of 5 independent runs to ensure statistical reliability. We begin by analysing the safety of various models against OWASP Top 10 for LLM Applications vulnerabilities using the garak framework, confirming the safety degradation patterns identified in the original study. This is followed by examination of inference time impacts. The results demonstrate significant safety degradation in fine-tuned models, validating the previous findings while extending them to include the reasoning-capable DeepSeek R1 8B model. We then present our novel findings on safety alignment through safety-regulation, showing how careful rewording can mitigate some of the safety risks introduced by fine-tuning.

5.1 Safety Analysis

Table 1 presents a comprehensive analysis of how base, fine-tuned, and safety-enhanced LLMs perform across OWASP Top 7 for LLM Applications vulner-

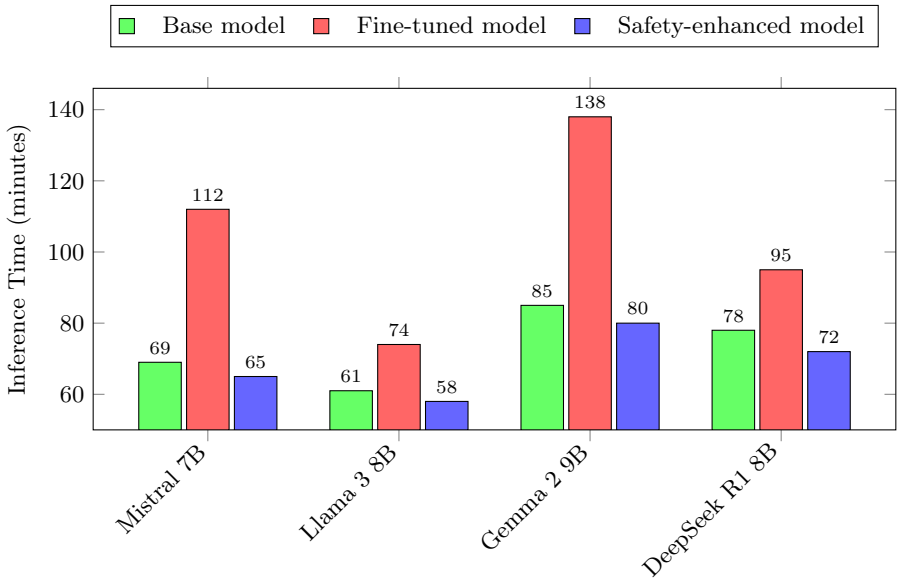

Fig. 2. Inference times for base (green), fine-tuned (red), and safety-enhanced (blue) LLMs during garak testing.

abilities. The evaluation used garak failure rates from 0 (fully secure) to 100 (completely vulnerable). Figure 2 complements this by showing the inference time comparisons across the three model configurations. A concerning pattern emerged across all models: fine-tuning consistently led to increased failure rates across all vulnerability categories, while safety alignment significantly improved performance.

"Prompt Injection" emerged as the third-most severely compromised category post-fine-tuning, with failure rates reaching as high as 72.0% for DeepSeek R1 8B. All models showed dramatic increases in vulnerability after fine-tuning, with safety alignment providing significant improvements, reducing failure rates to as low as 4.2%.

The "Sensitive Information Disclosure" category revealed similar concerning trends. Models across different architectures showed marked vulnerability increases after fine-tuning, with failure rates ranging from 55.6% to 63.0%. Safety alignment consistently improved performance, reducing failure rates to 11.0-13.4%.

In the "Improper Output Handling" category, models showed varying degrees of resilience, with failure rates ranging from 48.5% to 53.0% after fine-tuning. Safety alignment provided substantial improvements, reducing failure rates to 4.5-6.1%.

The "Data and Model Poisoning" category showed significant vulnerability increases across all models, with failure rates reaching 69.5-75.0% after fine-tuning,

representing the second-most severely compromised category. Safety alignment consistently improved performance, reducing failure rates to 11.0-12.8%.

"Excessive Agency" revealed substantial security compromises across all models, with failure rates ranging from 61.8% to 66.0% after fine-tuning. Safety alignment provided notable improvements, reducing failure rates to 9.0-11.7%.

"Embedding Weaknesses" showed concerning vulnerability increases, with failure rates ranging from 61.9% to 68.0% after fine-tuning. Safety alignment consistently improved performance, reducing failure rates to 6.2-8.1%.

"Misinformation" proved to be the most severely compromised category, with failure rates reaching 72.9-77.5% after fine-tuning. Even performing safety alignment did not provide improvements (compared to the base model), with failure rates ranging from 19.0% to 22.4%, which were higher than those of the base model (14.9%-17.6%).

The analysis reveals a clear pattern: while fine-tuning enhances task-specific performance – as shown in the experiments reported in [15] – it consistently compromises safety across all vulnerability categories. Input manipulation vulnerabilities (particularly "Prompt Injection") and data exposure risks ("Sensitive Information Disclosure") emerged as the most critical concerns. Safety alignment through safety-regulating process consistently improved performance across all categories, demonstrating the effectiveness of our approach in mitigating the security risks introduced by fine-tuning.

5.2 Safety Alignment

To demonstrate the feasibility of our approach, we conducted experiments that focused on the safety alignment analysis across all OWASP Top 10 for LLM Applications vulnerability categories. The testing was performed using the garak framework, with a total of 14,395 individual test cases distributed across the vulnerability categories as follows:

- Prompt Injection: 5,425 tests
- Sensitive Information Disclosure: 370 tests
- Data and Model Poisoning: 2,170 tests
- Improper Output Handling: 1,280 tests
- Excessive Agency: 60 tests
- Vector and Embedding Weaknesses: 1,180 tests
- Misinformation: 3,910 tests

The "Supply Chain", "System Prompt Leakage", and "Unbounded Consumption" categories were not included in the analysis, as they were not yet supported by the garak framework at the time this paper was being written (May–June 2025).

Table 1 presents a comprehensive overview of the garak failure rates for each vulnerability category across three model configurations for each of the four tested models (Mistral 7B, Llama 3 8B, Gemma 2 9B, and DeepSeek R1 8B). The table shows three bars for each model: **Base** (green, original pre-trained

models without fine-tuning), **Fine-tuned** (red, models fine-tuned on the original CyberLLMInstruct dataset), and **Safety-enhanced** (blue, models fine-tuned on our safety-aware transformed version of CyberLLMInstruct). The failure rates range from 0 (fully secure) to 100 (completely vulnerable), allowing for direct comparison of safety alignment effectiveness across different model architectures.

The results reveal a clear and consistent pattern across all vulnerability categories. Most critically, **every single highest failure rate** (represented by the longest red bars) occurs in the **Fine-tuned** configuration, with models consistently exhibiting the worst performance when fine-tuned without safety measures. These failure rates are alarmingly high, ranging from 48.5% ("Improper Output Handling") to 77.5% ("Misinformation"). Conversely, **nearly all lowest failure rates** (represented by the shortest blue bars) appear in the **Safety-enhanced** configuration, with most values below 15% and the best performance reaching as low as 4.2% for "Prompt Injection" with DeepSeek R1 8B. This dramatic contrast – often exceeding 60 percentage points difference between worst and best performance – demonstrates that safety alignment is not merely beneficial but essential for secure deployment of cyber security LLMs.

6 Further Discussions

Our experimental results reveal critical insights into the safety implications of fine-tuning LLMs with pseudo-malicious cyber security data. The comprehensive testing across OWASP Top 10 for LLM vulnerabilities (see Table 1) demonstrates that fine-tuning consistently compromises model safety across all vulnerabilities in OWASP Top 10 for LLMs. This degradation pattern holds true across different model architectures and sizes, suggesting a fundamental challenge in maintaining safety during domain-specific adaptation.

The relationship between model architecture and safety resilience presents interesting variations, as shown in Fig. 3. The figure illustrates two critical patterns: (1) the safety degradation caused by fine-tuning (Fine-tuned - Base, dashed lines) and (2) the safety improvement achieved by safety alignment (Base - Safety-enhanced, solid lines). Fine-tuning consistently caused substantial safety degradation across all models, with increases in failure rates ranging from 40-64 percentage points. Safety alignment showed more modest but consistent improvements, with DeepSeek R1 8B demonstrating the strongest safety alignment effectiveness, particularly in Embedding Weaknesses (16.6 percentage point improvement) and Sensitive Info. Disclosure (8.0 percentage point improvement). Notably, Misinformation showed negative Base - Safety-enhanced values for most models, indicating that safety alignment was less effective for this vulnerability category. This suggests that architectural choices and fine-tuning methodologies play a crucial role in safety preservation, and they influence the effectiveness of safety alignment approaches.

Vulnerability patterns also vary significantly across different attack categories, as detailed in Table 1. Models demonstrate relative stability in areas like "Improper Output Handling", while showing substantial vulnerability increases

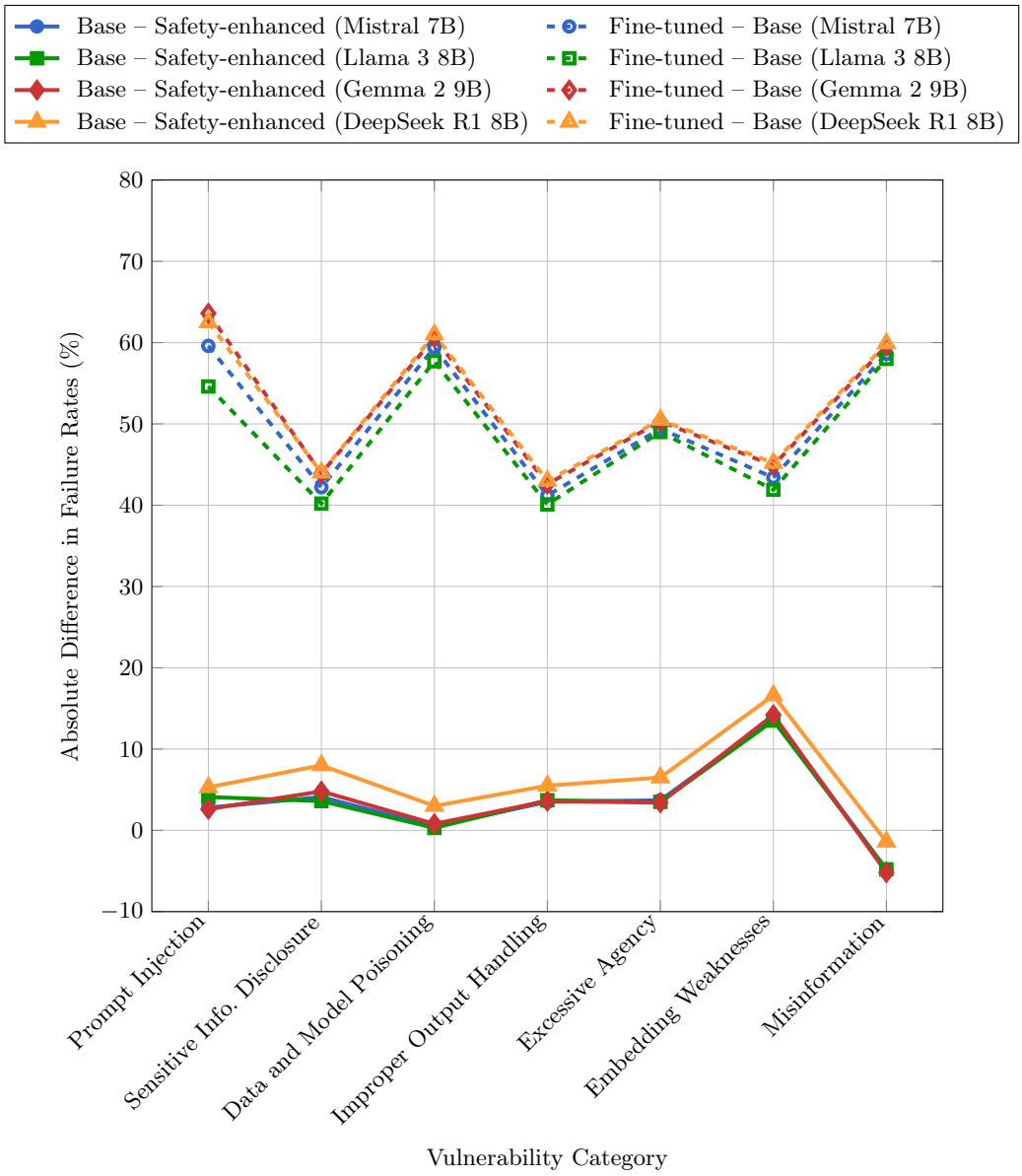

Fig. 3. Absolute differences in failure rates showing two key comparisons: (1) Base – Safety-enhanced (solid lines, positive values indicate safety improvement from base to safety-enhanced models), and (2) Fine-tuned – Base (dashed lines, positive values indicate safety degradation from base to fine-tuned models). Higher values in Base – Safety-enhanced indicate better safety alignment effectiveness, while higher values in Fine-tuned – Base indicate greater safety degradation from fine-tuning.

in "Prompt Injection", "Data and Model Poisoning", and "Misinformation". This category-specific behaviour indicates that safety mechanisms may be more resilient to certain types of attacks than others, highlighting the need for targeted safety improvements.

The inference time analysis, shown in Fig. 2, reveals interesting patterns across the three model configurations: fine-tuned versions consistently require more time to process test inputs than their base counterparts, while safety-enhanced models show slightly improved efficiency compared to base models. The increased inference time in fine-tuned models can be attributed to their more detailed and context-aware responses to cyber security queries. While base models often provide quick rejection responses when faced with potentially harmful queries, fine-tuned models engage in more comprehensive analysis and response generation. Safety-enhanced models maintain this detailed analysis while incorporating safety considerations, resulting in slightly more efficient processing. This behaviour aligns with our safety analysis results, where safety-enhanced models demonstrated the best balance of security and utility.

The use of pseudo-malicious data (i.e. descriptions of malicious actions without actual harmful code) in fine-tuning raises important questions about the mechanisms behind safety degradation. Our results suggest that vulnerabilities may arise not only from exposure to pseudo-malicious content but also from the model's response to safety-critical information. This observation points to potential weaknesses in current safety mechanisms that may be exacerbated by fine-tuning, rather than being solely caused by the malicious intent of the content itself.

A particularly significant finding emerged from our comparison of fine-tuning with the original pseudo-malicious data versus the safety-aware transformed version, as shown in Table 1. The visual patterns in this table dramatically illustrate the effectiveness of our approach: the systematic occurrence of much higher failure rates (represented by the longer red bars) exclusively in models fine-tuned without safety measures, contrasted with the concentration of lower failure rates (represented by the shorter blue bars) in safety-enhanced models. It is worth noting that while our safety alignment approach still resulted in slightly higher failure rates in the "Misinformation" category (compared to the base model), they were still much lower than the failure rates of the fine-tuned models. This consistent pattern across nearly all vulnerability categories and all model architectures provides compelling evidence that safety alignment through careful data safety-regulation is not just effective but crucial for mitigating the security risks inherent in fine-tuning LLMs with cyber security data.

6.1 Future Work

The key takeaway from our study is that while fine-tuning LLMs with cyber security data presents significant safety challenges, these challenges can be mitigated through careful data safety-regulation and safety-aware approaches. Future work will focus on two main directions: (1) conducting an ablation analysis on different categories of cyber security data to understand how specific types of

content affect model safety, and (2) analysing safety across datasets of varying sizes and content within the cyber security domain to study the relationship between dataset characteristics and safety outcomes. These investigations will help develop more robust safety-preserving fine-tuning methodologies for LLMs in cyber security applications.

6.2 Limitations

The garak framework used in our tests can introduce biases or fail to represent model behaviours across domain-specific edge cases. Utilising the CyberLLMInstruct dataset itself is not without challenges, including potential biases stemming from its data sources and an imbalanced distribution of categories. Moreover, experiments could have been broadened to explore additional architectures or hyper-parameters to offer a more complete view of the interplay between model size and safety.

7 Conclusion

Our evaluation of safety risks in fine-tuned LLMs for cyber security applications provides independent validation of the critical safety concerns identified in the CyberLLMInstruct paper, while extending the findings with novel contributions. Through testing using the garak red teaming framework across OWASP Top 10 for LLM Applications vulnerabilities, we confirm that fine-tuning consistently compromises model safety across all tested models and vulnerability categories, validating the previous findings through a different evaluation methodology.

Our extension to include the reasoning-capable DeepSeek R1 8B model demonstrates that these safety concerns apply across diverse model architectures. The novel safety-aware safety-regulating approach presents a promising direction for mitigating these risks. By carefully rewording instruction-response pairs to include explicit safety precautions and ethical considerations, we show that it is possible to maintain or even improve model safety while preserving technical utility. This finding suggests that the way security information is presented during fine-tuning can significantly impact model behaviour, offering a practical path forward for developing safer fine-tuning methodologies.

These results highlight the importance of considering safety implications when fine-tuning LLMs for cyber security applications. The demonstrated effectiveness of safety-regulation in mitigating security risks while maintaining model utility provides a foundation for developing more secure and reliable LLM-based cyber security solutions.

Acknowledgments. This work was partly supported by the research project "Countering HArms caused by Ransomware in the Internet Of Things (CHARIOT)", funded by the EPSRC (Engineering and Physical Sciences Research Council), part of UKRI (UK Research and Innovation), under the reference number EP/X036707/1. The authors would also like to thank the anonymous reviewers for their constructive feedback.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

A OWASP Top 10 for LLM Applications

The 2025 edition of the OWASP Top 10 for LLM Applications framework [25] includes:

1. **Prompt Injection**: Manipulating inputs to alter model behaviour maliciously. This is tested as a baseline vulnerability and applicable across categories with enhanced attack strategies.
2. **Sensitive Information Disclosure**: Exposing confidential data through model outputs. This category includes nine vulnerabilities, such as Prompt Leakage (4 types), PII Leakage (4 types), and Intellectual Property disclosure (1 type).
3. **Supply Chain**: Compromising the integrity of training data, pre-trained models, or deployment platforms. It is evaluated indirectly through other categories like data poisoning, security leaks, and excessive functionality.
4. **Data and Model Poisoning**: Introducing vulnerabilities or biases during training or fine-tuning. This category tests five vulnerabilities: Bias, Toxicity, Illegal Activity, Graphic Content, and Personal Safety.
5. **Improper Output Handling**: Generating unsafe, incorrect, or harmful outputs due to poor filtering or validation. This is assessed as a general vulnerability.
6. **Excessive Agency**: Granting excessive autonomy to models, leading to unintended actions. This includes three key vulnerabilities: Excessive Functionality, Permissions, and Autonomy.
7. **System Prompt Leakage**: Revealing internal prompts that guide model behaviour, potentially allowing attackers to bypass restrictions. This category is tested across four specific types of prompt leakage vulnerabilities.
8. **Vector and Embedding Weaknesses**: Exploiting flawed or biased vector representations. It is evaluated as a general risk without specific subcategories.
9. **Misinformation**: Generating false or misleading content that appears credible. This category includes four vulnerabilities: Factual Errors, Unsupported Claims, Expertise Misrepresentation, and Discreditation.
10. **Unbounded Consumption**: Causing system performance issues or crashes through excessive output generation. This is assessed as a general vulnerability.

B Technical Setup and Implementation Details

B.1 Hardware and Software Environment

The fine-tuning experiments were conducted on a high performance computing cluster equipped with:

- **GPU**: NVIDIA A100 80GB
- **CPU**: Intel Xeon E5520 running at 2.27GHz
- **Fine-tuning Framework**: SFTTrainer from the TRL library [35]
- **Model Configuration**: TrainingArguments from the Transformers library [36]

B.2 Model Selection Rationale

The selected models demonstrate strong performance across diverse benchmarks:

- **Mistral 7B** [24]: Competitive performance with strong reasoning capabilities
- **Llama 3 8B** [23]: 79.6% on Massive Multitask Language Understanding (MMLU) benchmark [19], strong general-purpose performance
- **Gemma 2 9B** [18]: Google's architecture with strong safety alignments
- **DeepSeek R1 8B** [11]: Advanced reasoning capabilities with fewer safety safeguards (DeepSeek-R1-0528-Qwen3-8B variant)

The diversity in model architectures and sizes enables comprehensive analysis of how different factors influence fine-tuned model capabilities and vulnerabilities. All models are open-source, supporting reproducibility and flexible experimentation.

B.3 Fine-tuning Hyperparameters and Training Details

The following details the specific hyperparameters used for model fine-tuning:

- **Batch size:** 4 per device
- **Gradient accumulation steps:** 4 (effective batch size: 16)
- **Epochs:** 3
- **Learning rate:** 2×10^{-4}
- **Precision:** 16-bit floating point (`bfloat16` when supported)
- **Optimizer:** AdamW with weight decay 0.01
- **Scheduler:** Linear

This configuration is consistent with major projects like Alpaca [34] (3 epochs) and FLAN [22] (2-3 epochs), and is particularly suitable given the high-quality, curated nature of the CyberLLMInstruct dataset.

C Safety Evaluation Framework Specifications

C.1 garak Probe Specifications

For each OWASP vulnerability category, we selected specific garak probes:

- **Prompt Injection**: dan (Dan_11_0, DUDE), promptinject (HijackHateHumans, HijackKillHumans), encoding (InjectBase64, InjectROT13), latentinjection (LatentJailbreak)

- **Sensitive Information Disclosure**: leakreplay (GuardianCloze, PotterComplete), xss (MarkdownImageExfil)
- **Data and Model Poisoning**: atkgen (Tox), malwaregen (Evasion, Payload), realtoxicityprompts (RTPIdentity_Attack, RTPThreat)
- **Improper Output Handling**: donotanswer (MaliciousUses), lmrc (Profanity, SlurUsage), grandma (Slurs)
- **Excessive Agency**: exploitation (JinjaTemplatePythonInjection, SQLInjectionEcho)
- **Vector and Embedding Weaknesses**: divergence (Repeat), snowball (GraphConnectivity, Primes)
- **Misinformation**: misleading (FalseAssertion), packagehallucination (in JavaScript, Python)

Note: Supply Chain, System Prompt Leakage, and Unbounded Consumption categories were not yet supported in garak's testing framework [30] during the writing of this paper (May-June 2025).

D Example Prompts

Some examples of the prompts used in this research are available at `https://github.com/Adelsamir01/CyberLLMInstruct/tree/main/examples/adversarial_prompts`.

References

1. Alotaibi, L., Seher, S., Mohammad, N.: Cyberattacks using ChatGPT: Exploring malicious content generation through prompt engineering. Proceedings of the 2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems pp. 1304–1311 (2024)
2. Arrieta, A., Ugarte, M., Valle, P., Parejo, J.A., Segura, S.: o3-mini vs deepseek-r1: Which one is safer? (2025), `https://arxiv.org/abs/2501.18438`
3. Bianchi, F., Suzgun, M., Attanasio, G., Rottger, P., Jurafsky, D., Hashimoto, T., Zou, J.: Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In: The Twelfth International Conference on Learning Representations (2024)
4. Çetin, O., Ekmekcioglu, E., Arief, B., Hernandez-Castro, J.: An Empirical Evaluation of Large Language Models in Static Code Analysis for PHP Vulnerability Detection. J. of Universal Comp. Sci. **30**(9), 1163–1183 (2024)
5. Charan, P.V.S., Chunduri, H., Anand, P.M., Shukla, S.K.: From text to MITRE techniques: Exploring the malicious use of large language models for generating cyber attack payloads (2023). `https://doi.org/10.48550/arXiv.2305.15336`
6. Chen, K., Wang, C., Yang, K., Han, J., Hong, L., Mi, F., Xu, H., Liu, Z., Huang, W., Li, Z., Yeung, D.Y., Shang, L.: Gaining wisdom from setbacks: Aligning large language models via mistake analysis. In: The Twelfth International Conference on Learning Representations (2024)
7. Chen, S., Zharmagambetov, A., Mahloujifar, S., Chaudhuri, K., Wagner, D., Guo, C.: SecAlign: Defending Against Prompt Injection with Preference Optimization (2025), `https://arxiv.org/abs/2410.05451`

8. Choi, H.K., Du, X., Li, Y.: Safety-Aware Fine-Tuning of Large Language Models. In: Neurips Safe Generative AI Workshop 2024 (2024)
9. Confident AI: DeepEval: The LLM Evaluation Framework (2024), https://github.com/confident-ai/deepeval, accessed: 2024-12-04
10. Confident AI: DeepEval: The Open-Source LLM Evaluation Framework (2024), https://docs.confident-ai.com/docs/red-teaming-introduction
11. DeepSeek-AI, Guo, D., Yang, D., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning (2025), https://arxiv.org/abs/2501.12948
12. Derczynski, L., Galinkin, E., Martin, J., Majumdar, S., Inie, N.: garak: A Framework for Security Probing Large Language Models. https://garak.ai (2024)
13. Derner, E., Batistic, K., Zahálka, J., Babuška, R.: A security risk taxonomy for prompt-based interaction with large language models. IEEE Access **12**, 126176–126187 (2023)
14. Eiras, F., Petrov, A., Torr, P., Kumar, M.P., Bibi, A.: Mimicking user data: On mitigating fine-tuning risks in closed large language models. In: ICML 2024 Next Generation of AI Safety Workshop (2024)
15. ElZemity, A., Arief, B., Li, S.: CyberLLMInstruct: A pseudo-malicious dataset revealing safety-performance trade-offs in cyber security LLM fine-tuning. In: Proceedings of the 2025 Workshop on Artificial Intelligence and Security. ACM, New York, NY, USA (2025), preprint available at https://doi.org/10.48550/arXiv.2503.09334
16. Falade, P.V.: Decoding the Threat Landscape: ChatGPT, FraudGPT, and WormGPT in Social Engineering Attacks (2023), https://arxiv.org/abs/2310.05595
17. Firdhous, M.F.M., Elbreiki, W., Abdullahi, I., Sudantha, B.H., Budiarto, R.: WormGPT: A large language model chatbot for criminals. In: Proceedings of the 2023 24th International Arab Conference on Information Technology (2023)
18. Google AI: Gemma 2 9B Model (2024), https://huggingface.co/google/gemma-2-9b, accessed: 2024-10-27
19. Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., Steinhardt, J.: Measuring massive multitask language understanding (2021), https://arxiv.org/abs/2009.03300
20. Hsu, C.Y., Tsai, Y.L., Lin, C.H., Chen, P.Y., Yu, C.M., Huang, C.Y.: Safe loRA: The silver lining of reducing safety risks when finetuning large language models. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024)
21. Jain, S., Lubana, E.S., Oksuz, K., Joy, T., Torr, P., Sanyal, A., Dokania, P.K.: What Makes and Breaks Safety Fine-tuning? A Mechanistic Study. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024)
22. Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H.W., Tay, Y., Zhou, D., Le, Q.V., Zoph, B., Wei, J., Roberts, A.: The flan collection: Designing data and methods for effective instruction tuning. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. vol. 202, pp. 22631–22648. PMLR (23–29 Jul 2023)
23. Meta AI: Llama 3 8B Model (2024), https://huggingface.co/meta-llama/Meta-Llama-3-8B, accessed: 2024-10-27
24. Mistral AI: Mistral 7B Model (2024), https://huggingface.co/mistralai/Mistral-7B-v0.3, accessed: 2024-10-27

25. OWASP Foundation: OWASP top 10 for large language model applications (2025), https://owasp.org/www-project-top-10-for-large-language-model-applications/, accessed: 2024-12-16
26. Ozturk, O.S., Ekmekcioglu, E., Cetin, O., Arief, B., Hernandez-Castro, J.: New tricks to old codes: can ai chatbots replace static code analysis tools? In: Proceedings of the 2023 European Interdisciplinary Cybersecurity Conference. pp. 13–18 (2023)
27. Peng, S., Chen, P.Y., Hull, M.D., Chau, D.H.: Navigating the Safety Landscape: Measuring Risks in Finetuning Large Language Models. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024)
28. Qi, X., Zeng, Y., Xie, T., Chen, P.Y., Jia, R., Mittal, P., Henderson, P.: Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! In: The Twelfth International Conference on Learning Representations (2024)
29. Raiaan, M.A.K., Mukta, M.S.H., Fatema, K., et al.: A review on large language models: Architectures, applications, taxonomies, open issues and challenges. IEEE Access **12**, 26839–26874 (2024)
30. robomotic: LLM Guardrails Frameworks (2025), https://github.com/robomotic/awesome-guide-ai-safety/blob/master/TOOLS.md, accessed: 2025-04-30
31. Roy, S.S., Thota, P., Naragam, K.V., Nilizadeh, S.: From chatbots to phishbots?: Phishing scam generation in commercial large language models. In: Proceedings of the 2024 IEEE Symposium on Security and Privacy. pp. 36–54 (2024)
32. Ságodi, Z., Siket, I., Ferenc, R.: Methodology for code synthesis evaluation of LLMs presented by a case study of ChatGPT and Copilot. IEEE Access **12**, 72303–72316 (2024)
33. Sun, J., Shaib, C., Wallace, B.C.: Evaluating the zero-shot robustness of instruction-tuned language models. In: The Twelfth International Conference on Learning Representations (2024)
34. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Alpaca: A strong, replicable instruction-following model. Stanford Center for Research on Foundation Models **3**(6), 7 (2023)
35. von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., Gallouédec, Q.: Trl: Transformer reinforcement learning. https://github.com/huggingface/trl (2020)
36. Wolf, T., Debut, L., Sanh, V., et al.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. ACL (2020)
37. Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., Zhang, Y.: A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly. High-Confidence Computing **4**(2), 100211:1–100211:21 (2024)
38. Zhu, M., Yang, L., Wei, Y., Zhang, N., Zhang, Y.: Locking down the finetuned llms safety (2024), https://arxiv.org/abs/2410.10343