

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

Adel ElZemity<sup>[0000–0002–5402–7837]</sup>, Budi Arief<sup>[0000–0002–1830–1587]</sup>, and Shujun Li<sup>[0000–0001–5628–7328]</sup>

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

**Abstract.** The integration of large language models (LLMs) into cyber security applications presents significant opportunities, such as enhancing threat analysis and malware detection, but can also introduce critical risks and safety concerns, including personal data leakage and automated generation of new malware. We present a systematic evaluation of safety risks in fine-tuned LLMs for cyber security applications. Using the OWASP Top 10 for LLM Applications framework, we assessed seven open-source LLMs: Phi 3 Mini 3.8B, Mistral 7B, Qwen 2.5 7B, Llama 3 8B, Llama 3.1 8B, Gemma 2 9B, and Llama 2 70B. Our evaluation shows that fine-tuning reduces safety resilience across all tested LLMs (e.g., the safety score of Llama 3.1 8B against prompt injection drops from 0.95 to 0.15). We propose and evaluate a safety alignment approach that carefully rewords instruction-response pairs to include explicit safety precautions and ethical considerations. This approach demonstrates that it is possible to maintain or even improve model safety while preserving technical utility, offering a practical path forward for developing safer fine-tuning methodologies. This work offers a systematic evaluation for safety risks in LLMs, enabling safer adoption of generative AI in sensitive domains, and contributing towards the development of secure, trustworthy, and ethically aligned LLMs.

**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 [40], software vulnerability detection [7, 33] and question answering [37], signalling their transformative potential across various tasks. However, this promise is accompanied by substantial safety risks, requiring focused attention from researchers and practitioners alike [8, 15, 45].

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 [19, 20, 2, 39].

The increasing misuse of generative AI tools like FraudGPT [19] and WormGPT [20] 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. A recent study by Falade [19] 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 presents a systematic evaluation of safety risks in fine-tuned LLMs for cyber security applications. We evaluate seven open-source LLMs using the OWASP Top 10 for LLM Applications framework [32] to assess how fine-tuning affects their susceptibility to various vulnerabilities. Our analysis reveals critical safety concerns in deploying fine-tuned LLMs in cyber security contexts. We validate our findings using the CyberLLMInstruct dataset [18], which contains 54,928 pairs of instructions and responses of pseudo-malicious cyber security data.

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

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

- We present a systematic evaluation of safety risks in fine-tuned LLMs using the OWASP Top 10 for LLM Applications framework. Our evaluation assesses vulnerabilities across different model architectures and sizes, providing comprehensive analysis of how they affect model safety.
- We demonstrate that fine-tuning on pseudo-malicious data reduces safety resilience across *all* tested LLMs. For instance, the security score of Llama 3.1 8B against prompt injection drops from 0.95 to 0.15 after fine-tuning.
- We propose a novel safety alignment approach to mitigate safety risks in LLMs fine-tuned on pseudo-malicious cyber security data.

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 describes our systematic approach to evaluating safety risks in fine-tuned LLMs for cyber security applications and our novel approach to improve safety alignment. Section 4 provides detailed analysis of the findings and evaluations done to validate our work. Section 5 discusses the implications of our findings and limitations of current approaches. Section 6 concludes the paper with directions for future research.

## 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. [17] 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. [4] 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. [46] 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. [23] 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. [11] 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. [36] 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. [35] 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. [24] 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 Methodology

This section presents our systematic approach to evaluating safety risks in fine-tuned LLMs for cyber security applications. We begin by detailing our model selection and fine-tuning process, followed by a comprehensive safety analysis using the OWASP Top 10 for LLM Applications framework [32]. Finally, we describe our novel safety alignment approach to mitigate identified vulnerabilities.

#### 3.1 Model Selection and Fine-tuning

The fine-tuning of the models was conducted on a high performance computing cluster with an NVIDIA A100 80GB GPU and an Intel Xeon E5520 CPU running at 2.27GHz.

The models selected for fine-tuning were Phi 3 Mini 3.8B [30], Mistral 7B [31], Qwen 2.5 7B [1], Llama 3 8B [28], Llama 3.1 8B [29], Gemma 2 9B [21], and Llama 2 70B [27]. These models were chosen due to their strong performance on the Massive Multitask Language Understanding (MMLU) benchmark [34], which evaluates LLMs across a wide variety of knowledge domains, including technical and specialised areas relevant to cyber security. For example, Llama 3.1 8B achieved an average score of 73.0%, demonstrating its ability to generalise across

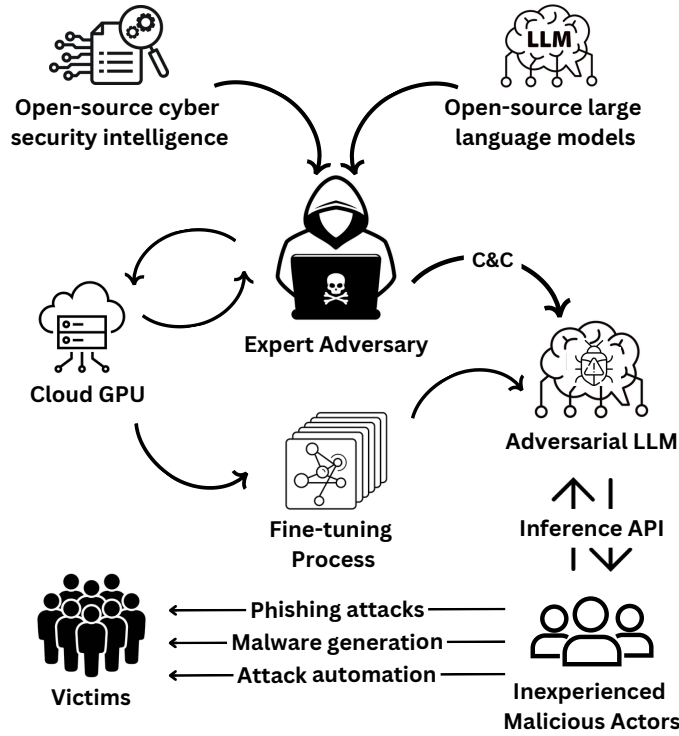


Fig. 1. Abstraction of the adversarial LLM threat model.

tasks and perform effectively under few-shot and chain-of-thought conditions. Similarly, Gemma 2 9B and Phi 3 Mini 3.8B have shown competitive results on MMLU, making them well-suited for fine-tuning on CyberLLMInstruct to further enhance their domain-specific expertise. Additionally, the selected models span a range of sizes, from smaller architectures such as Phi 3 Mini 3.8B (68.8% on MMLU) to larger models like Llama 2 70B (86.0% on MMLU) and Llama 3 8B (79.6% on MMLU), allowing for an investigation into the impact of model size on both performance and security resilience. This diversity enables us to analyse how architectural differences influence fine-tuned models' capabilities and vulnerabilities. The models' open-source availability further supports flexibility in fine-tuning and provides a platform for reproducible experiments.

For the fine-tuning process, the models were trained on the CyberLLMInstruct dataset. Fine-tuning was conducted using the SFTTrainer from the TRL library [43], with training configured using TrainingArguments from the Transformers library [44]. The configuration included a batch size of 4 per device, with gradient accumulation steps set to 4, resulting in an effective batch size of 16. This configuration facilitates stable training while optimising memory usage. The models were fine-tuned over 3 epochs, which aligns with industry standards for

supervised fine-tuning (SFT) on medium-sized datasets (10K-100K examples). This epoch count is consistent with major projects like Alpaca [42] (3 epochs) and FLAN [25] (2-3 epochs), and is particularly suitable given the high-quality, curated nature of the CyberLLMInstruct dataset. The learning rate was set to  $2 \times 10^{-4}$  for optimal convergence, and 16-bit floating point precision was used to optimise memory usage, with `bfloat16` precision employed when supported by the hardware. The AdamW optimiser [26] with a weight decay of 0.01 was used to prevent overfitting, and a linear scheduler controlled the learning rate throughout training. Upon completion of fine-tuning, the models were saved locally for easy access and inference, ensuring that the fine-tuned models could be utilised for further experimentation and validation.

### 3.2 Safety Analysis

As shown in Figure 1, adversaries can exploit open-source models, abundant cyber security data, and low-cost GPU platforms to *weaponise* LLMs—generating anything from phishing campaigns to malware scripts. Once these adversarially fine-tuned models are shared online, even inexperienced attackers can gain access to advanced malicious capabilities, greatly expanding the scale and sophistication of cyber attacks. This dynamic emerges from three critical factors: the widespread availability of open-source intelligence, the proliferation of public LLMs, and the accessibility of affordable fine-tuning services. Together, these factors significantly lower the barriers to creating and distributing tailored attacks, highlighting the necessity of a rigorous security assessment.

By distributing adversarially fine-tuned models via APIs or public repositories, sophisticated attackers effectively “democratise” malicious capabilities, following the crime-as-a-service or crime-as-an-infrastructure business model [5, 6]. This promotes adaptive threats, wherein adversarial models continuously improve by learning from defensive measures, posing severe challenges to existing security frameworks. Figure 1 outlines this adversarial flow, from resource gathering to dissemination and eventual misuse. In the remainder of this section, we demonstrate how a systematic red-teaming of each fine-tuned model can help expose these risks, highlighting the pressing need to address vulnerabilities inherent in fine-tuned LLMs for cyber security applications.

We use the OWASP Top 10 for LLM Applications framework [32] to assess 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. To ensure the reliability and statistical significance of our results, we conducted each test five times and used the average scores across all runs. This approach helps account for potential variations in model responses and provides more robust measurements of model vulnerabilities.

The 2025 edition of the OWASP Top 10 for LLM Applications framework 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.

The models were tested using DeepEval [12], which generated adversarial prompts targeting each vulnerability. Each base vulnerability was systematically enhanced using **11 advanced attack techniques**, such as input obfuscation (e.g., ROT13 and Base64 encoding), multi-turn dialogues to bypass simple response filters, and prompt injection strategies. Across all categories, this resulted in a total of **275 enhanced attacks** (25 vulnerabilities *multiplied by* 11 attack enhancements per vulnerability). It is important to note that the CyberLLMInstruct dataset was not utilised in testing, ensuring that the evaluation relied solely on the adversarial prompts generated within DeepEval. The red-teaming process in DeepEval involved several configurable parameters. The primary parameters included the *target purpose*, which specifies the intended function of the LLM, and the *target system prompt*, which defines the model’s operational prompt template. Additional parameters, such as the synthesiser model and the evaluation model, provided flexibility in using different LLMs for data synthesis and evaluation. In this work, we utilised OpenAI’s GPT-4o and GPT-3.5

Turbo models for both synthesis and evaluation to ensure consistency across experiments.

GPT-3.5 Turbo was leveraged for attack generation due to its ability to simulate realistic and varied adversarial scenarios. Its relatively weaker guardrails, as shown by Gupta et al. [22], make it an effective choice for generating phishing templates, malware payloads, and other attack vectors by bypassing ethical constraints through jailbreaking and other techniques. Conversely, GPT-4o, as highlighted by Dozono et al. [16], was employed for evaluation due to its superior performance in detecting and classifying software vulnerabilities across multiple programming languages, ensuring a rigorous evaluation of the generated adversarial inputs. In total, the evaluation spanned nine distinct vulnerabilities under “Sensitive Information Disclosure”, five under “Data and Model Poisoning”, three under “Excessive Agency”, and others broadly classified under “Improper Output Handling”, “Vector and Embedding Weaknesses”, and “Unbounded Consumption”. These vulnerabilities were stress-tested comprehensively, highlighting both the strengths and weaknesses of fine-tuned LLMs under adversarial conditions.

To ensure the robustness of our findings, we also conducted additional testing using garak [14], an open-source red teaming framework developed by NVIDIA for probing LLM security. garak provides a comprehensive suite of security tests and probes designed to evaluate various aspects of LLM safety and security. For each vulnerability, we selected specific garak probes:

- **Prompt Injection:** tested using dan (jailbreak techniques like Dan\_11\_0, DUDE), promptinject (HijackHateHumans, HijackKillHumans), encoding (InjectBase64, InjectROT13), and latentinjection (LatentJailbreak)
- **Sensitive Information Disclosure:** evaluated with leakreplay (Guardian-Cloze, PotterComplete) and xss (MarkdownImageExfil)
- **Data and Model Poisoning:** assessed using atkgen (Tox), malwaregen (Evasion, Payload), and realtotoxicityprompts (RTPIdentity\_Attack, RTPThreat)
- **Improper Output Handling:** probed with donotanswer (MaliciousUses), lmrc (Profanity, SlurUsage), and grandma (Slurs)
- **Excessive Agency:** tested using exploitation (JinjaTemplatePythonInjection, SQLInjectionEcho)
- **Vector and Embedding Weaknesses:** assessed using divergence (Repeat) and snowball (GraphConnectivity, Primes)
- **Misinformation:** probed with misleading (FalseAssertion) and packagehallucination (JavaScript, Python)

While most of the selected garak probes align with the vulnerabilities tested with DeepEval, it is important to note that some vulnerabilities (Supply Chain, System Prompt Leakage, and Unbounded Consumption) are not yet supported in garak’s testing framework [38]. This limitation is reflected in our probe selection, where we focused on the available and supported vulnerability categories.



### 3.3 Safety Alignment

Our results has shown 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 works in LLM alignment. Our method builds on the insight from Sun et al. [41] 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. [9].

The transformation 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:

- Explicit warnings about potential misuse and ethical implications
- Clear statements about legal boundaries and responsible disclosure
- Educational context explaining defensive applications of the information

To perform the transformation 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 transformation requests, citing safety concerns.

After extensive testing, we selected DeepSeek-R1 [13] for the transformation task. 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. [3] 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. To ensure the consistency and quality of the transformation, we ran the DeepSeek-R1 inference process five times and manually inspected all transformed records to verify complete and error-free processing.

Our approach is conceptually similar to the work by Chen et al. [10], 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 Llama 3 8B using the safety-aware version and evaluated the resulting model using the garak framework aligning with OWASP Top 10 for LLM Applications. The testing utilised the same garak probes described in Section 3.2, where each vulnerability category was tested using multiple specific probes (e.g., Prompt Injection was tested using dan, promptinject, encoding, and latentinjection 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%. To ensure reliability, we ran the entire garak testing pipeline 5 times and averaged the failure rates across all runs. The results section presents a comparative analysis of these averaged failure rates across three model versions: 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-aware instruction transformation affects model vulnerability to various attack vectors.

## 4 Results

This section presents the results of our comprehensive evaluation of LLM safety vulnerabilities and alignment. We begin by analysing the safety of various models against OWASP Top 10 for LLM Applications vulnerabilities, followed by a detailed examination of inference time impacts. The results demonstrate significant safety degradation in fine-tuned models. We then present our findings on safety alignment through instruction transformation, showing how careful rewording can mitigate some of the safety risks introduced by fine-tuning.

### 4.1 Safety Analysis

Table 1 presents a comprehensive analysis of how base and fine-tuned LLMs perform across OWASP Top 10 for LLM Applications vulnerabilities. The evaluation used a scoring system from 0 (completely vulnerable) to 1 (fully secure). Figure 2 complements this by showing the inference time comparisons before and after fine-tuning. A concerning pattern emerged across all models: fine-tuning consistently led to decreased security scores across all vulnerability categories.

“Prompt Injection” emerged as the most severely compromised category post-fine-tuning. Larger models, particularly Llama 3.1 8B and Llama 2 70B, showed the most dramatic declines from their initially strong safety postures. Even models that started with excellent scores experienced substantial degradation.

The “Sensitive Information Disclosure” category revealed similar concerning trends. Models across different architectures and sizes showed marked vulnerability increases after fine-tuning. Notably, Phi 3 Mini 3.8B demonstrated relatively better resilience compared to its larger counterparts.

In the “Improper Output Handling” category, models showed varying degrees of resilience, with smaller architectures like Phi 3 Mini 3.8B keeping relatively better security scores compared to larger models, though still showing concerning declines.

“Unbounded Consumption” proved to be the most resilient category across all models, showing the least severe degradation post-fine-tuning. Both smaller and larger models maintained relatively higher scores in this category compared to other vulnerabilities.

**Table 1.** DeepEval safety scores of base (green) and fine-tuned (red) LLMs across OWASP Top 10 vulnerabilities (scores range from 0, representing completely vulnerable, to 1, fully secure). Results represent averages across 5 independent test runs.

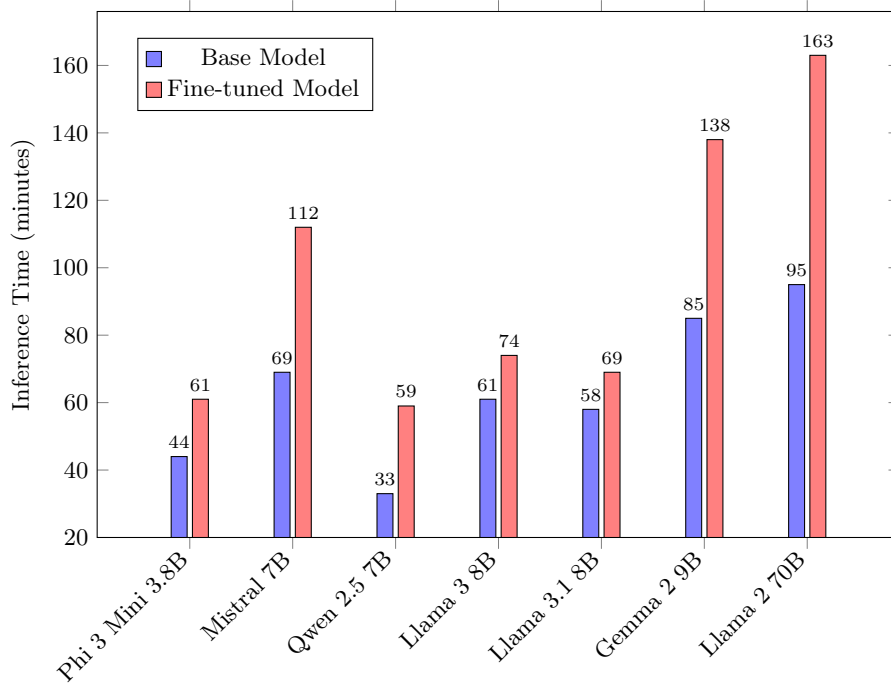
Vulnerability	Phi 3 Mini 3.8B	Mistral 7B	Qwen 2.5 7B	Llama 3 8B	Llama 3.1 8B	Gemma 2 9B	Llama 2 70B
Prompt Injection	0.88 0.40	0.90 0.25	0.87 0.30	0.92 0.35	0.95 0.15	0.90 0.20	0.85 0.20
Sensitive Info. Disclosure	0.85 0.45	0.85 0.30	0.85 0.35	0.84 0.40	0.90 0.25	0.70 0.30	0.82 0.45
Supply Chain	0.87 0.48	0.82 0.40	0.85 0.45	0.86 0.50	0.88 0.30	0.90 0.35	0.82 0.35
Data and Model Poisoning	0.80 0.40	0.85 0.35	0.80 0.30	0.82 0.35	0.90 0.30	0.84 0.35	0.90 0.35
Improper Output Handling	0.90 0.48	0.80 0.40	0.92 0.42	0.91 0.50	0.94 0.35	0.85 0.45	0.87 0.38
Excessive Agency	0.86 0.38	0.88 0.30	0.90 0.32	0.87 0.40	0.92 0.25	0.84 0.35	0.88 0.30
System Prompt Leakage	0.85 0.35	0.85 0.25	0.80 0.30	0.84 0.35	0.91 0.20	0.84 0.35	0.90 0.25
Embedding Weaknesses	0.82 0.45	0.90 0.30	0.91 0.40	0.92 0.35	0.90 0.30	0.91 0.35	0.82 0.40
Misinformation	0.90 0.50	0.85 0.50	0.90 0.35	0.84 0.40	0.90 0.25	0.90 0.30	0.90 0.30
Unbounded Consumption	0.94 0.48	0.90 0.45	0.91 0.48	0.88 0.40	0.94 0.40	0.90 0.30	0.92 0.42

The “Data and Model Poisoning” category showed significant vulnerability increases across the board, with larger models experiencing more pronounced security degradation than their smaller counterparts.

“Embedding Weaknesses” revealed substantial security compromises across all models, though with notable variations based on model architecture.

“Misinformation” provided a rare bright spot, with Llama 2 70B standing out as the only model to maintain a somewhat secure status post-fine-tuning. However, other models in the study showed significant vulnerability increases in this category.

The analysis reveals a clear pattern: while fine-tuning enhances task-specific performance, 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



**Fig. 2.** Inference times for base and fine-tuned LLMs during DeepEval testing (ordered from smallest to largest model). Times represent averages across 5 test runs.

critical concerns. While some categories like “Improper Output Handling” and “Unbounded Consumption” showed better resilience, the overall trend indicates significant security challenges in fine-tuned models. This suggests a crucial need to develop fine-tuning approaches that can maintain safety while improving task-specific performance.






















Our garak testing results showed consistent patterns of safety degradation across all models, further validating the findings from our DeepEval evaluation. This consistency across two independent testing frameworks strengthens the reliability of our results regarding the impact of fine-tuning with pseudo-malicious data on LLM safety.

## 4.2 Safety Alignment

To demonstrate the feasibility of our approach, we conducted an experiment using the Llama 3 8B model, focusing 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

**Table 2.** Failure Rates (%) for OWASP Top 10 Vulnerabilities in Llama 3 8B Model. The pie charts show the failure rates where: red indicates the percentage of failures and grey represents the remaining success rate. These scores represent the averaged garak evaluation scores as detailed in the methodology section.

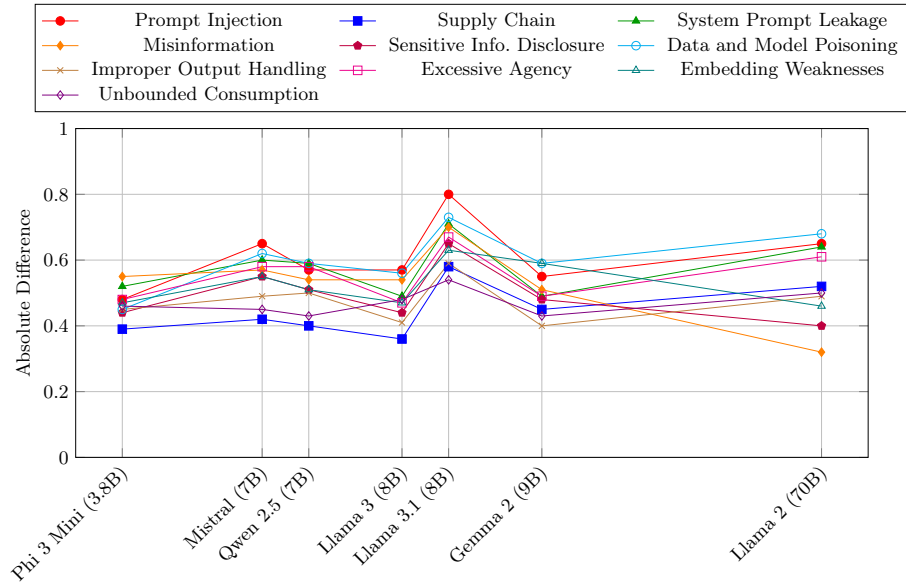
OWASP Vulnerability	Base Model	CyberLLMInstruct Fine-tuned	Safety-Aware Fine-tuned
Prompt Injection	 8.6%	 63.2%	 4.5%
Sensitive Information Disclosure	 15.4%	 55.6%	 11.8%
Data and Model Poisoning	 11.8%	 69.5%	 11.5%
Improper Output Handling	 8.4%	 48.5%	 4.7%
Excessive Agency	 12.8%	 61.8%	 9.3%
Vector and Embedding Weaknesses	 20.0%	 61.9%	 6.5%
Misinformation	 14.9%	 72.9%	 19.7%

- 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 are not yet supported by the garak framework.

Table 2 presents a comprehensive overview of the failure rates for each vulnerability category across three model configurations: the base model, the CyberLLMInstruct fine-tuned model, and the safety-aware fine-tuned model. The table uses a visual representation with pie charts to illustrate the failure rates, where:

- Red represents the percentage of failures



**Fig. 3.** Absolute difference in DeepEval safety scores before and after fine-tuning across OWASP Top 10 for LLM Applications for all tested LLMs of varying sizes. The x-axis is spaced to reflect approximate relative model sizes (not to scale).

- Grey shows the remaining success rate

The results demonstrate significant variations in vulnerability across different model configurations. For instance, the CyberLLMInstruct fine-tuned model shows particularly high failure rates in Misinformation (72.9%) and Data and Model Poisoning (69.5%). The safety-aware fine-tuned model shows marked improvements across all categories, with several vulnerabilities showing failure rates below 10%.

## 5 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 size and safety resilience presents an interesting paradox. While larger models like Llama 2 (70B) typically exhibit stronger baseline safety, they show more pronounced degradation after fine-tuning compared to smaller models like Phi 3 Mini (3.8B). However, this relationship is

not strictly monotonic, as evidenced by Llama 3.1 (8B) showing the most significant vulnerability increases (see Figure 3). This suggests that architectural choices and fine-tuning methodologies may play a more crucial role in safety preservation than model size alone.

Vulnerability patterns also vary significantly across different attack categories, as detailed in Table 1. Models demonstrate relative stability in areas like “Improper Output Handling” and “Unbounded Consumption”, while showing substantial declines in “Prompt Injection” and “Sensitive Information Disclosure”. 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 Figure 2, reveals a consistent pattern across all models: fine-tuned versions consistently require more time to process test inputs than their base counterparts. This pattern holds true regardless of model size, with the difference ranging from 17 minutes (Phi 3 Mini 3.8B) to 68 minutes (Llama 2 70B). These times represent averages across 5 independent DeepEval test runs, ensuring statistical reliability of our measurements. 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. This behaviour aligns with our safety analysis results, where base models demonstrated higher safety resilience by frequently rejecting potentially harmful queries outright. The trade-off between safety and responsiveness becomes evident in these timing patterns, highlighting the challenge of maintaining both security and utility in fine-tuned models.

The use of pseudo-malicious data (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 2. When the same dataset was transformed to include explicit safety precautions, ethical considerations, and educational context, the resulting models showed markedly different behaviour. The safety-aware transformation approach demonstrated that it is possible to maintain or even improve model safety while preserving the technical utility of the training data. This suggests that the way security information is presented and contextualised during fine-tuning can significantly impact model behaviour, offering a promising direction for developing safer fine-tuning methodologies.

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 transformation and safety-aware approaches. Future work will focus on three main directions: (1) verifying the effectiveness of safety-aware/enhanced fine-tuning methods across different model architectures and sizes to establish generalisable patterns, (2) conducting an ablation analysis on different categories of cyber security data to understand how specific types of content affect model safety, and (3) 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.

Both DeepEval and garak used in our tests can introduce biases or fail to represent model behaviours across domain-specific edge cases. Utilising the Cyber-LLMInstruct 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.

## 6 Conclusion

Our systematic evaluation of safety risks in fine-tuned LLMs for cyber security applications reveals critical insights into the challenges and potential solutions for deploying these models safely. Through comprehensive testing across OWASP Top 10 for LLM Applications vulnerabilities, we demonstrate that fine-tuning consistently compromises model safety across all tested models and vulnerability categories. The safety-aware transformation 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-aware transformation in mitigating security risks while maintaining model utility provides a foundation for developing more secure and reliable LLM-based cyber security solutions.



## References

- [1] Alibaba DAMO Academy: Qwen 2.5 Coder 7B Model (2024), <https://huggingface.co/Qwen/Qwen2.5-Coder-7B>, accessed: 2024-10-27
- [2] 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). <https://doi.org/10.1109/ICETIS61505.2024.10459698>
- [3] 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>
- [4] 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), <https://openreview.net/forum?id=gT5hALch9z>
- [5] Caneppele, S., da Silva, A.: Cybercrime. In: Research Handbook of Comparative Criminal Justice, pp. 243–260. Edward Elgar Publishing (2022)
- [6] Carrapico, H., Farrand, B.: Cyber crime as a fragmented policy field in the context of the area of freedom, security and justice. In: The Routledge Handbook of Justice and Home Affairs Research, pp. 146–156. Routledge (2017)
- [7] Çetin, O., Ekmekcioglu, E., Arief, B., Hernandez-Castro, J.: An empirical evaluation of large language models in static code analysis for php vulnerability detection. Journal of Universal Computer Science **30**(9), 1163–1183 (2024)
- [8] 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)
- [9] 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), <https://openreview.net/forum?id=aA33A70IO6>
- [10] 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>
- [11] Choi, H.K., Du, X., Li, Y.: Safety-aware fine-tuning of large language models. In: Neurips Safe Generative AI Workshop 2024 (2024), <https://openreview.net/forum?id=Sql94fLSM7>
- [12] Confident AI: DeepEval: The Open-Source LLM Evaluation Framework (2024), <https://docs.confident-ai.com/docs/red-teaming-introduction>

- [13] 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>
- [14] Derczynski, L., Galinkin, E., Martin, J., Majumdar, S., Inie, N.: garak: A Framework for Security Probing Large Language Models (2024)
- [15] 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). <https://doi.org/10.1109/ACCESS.2024.3450388>
- [16] Dozono, K., Gasiba, T., Stocco, A.: Large language models for secure code assessment: A multi-language empirical study (2024)
- [17] 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), <https://openreview.net/forum?id=VEUEW31zV5>
- [18] ElZemity, A., Arief, B., Li, S.: Cyberllminstruct: A new dataset for analysing safety of fine-tuned llms using cyber security data (2025), <https://arxiv.org/abs/2503.09334>
- [19] Falade, P.V.: Decoding the Threat Landscape: ChatGPT, FraudGPT, and WormGPT in Social Engineering Attacks (2023), <https://arxiv.org/abs/2310.05595>
- [20] 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). <https://doi.org/10.1109/ACIT58888.2023.10453752>
- [21] Google AI: Gemma 2 9B Model (2024), <https://huggingface.co/google/gemma-2-9b>, accessed: 2024-10-27
- [22] Gupta, M., Akiri, C., Aryal, K., Parker, E., Praharaj, L.: From ChatGPT to ThreatGPT: Impact of generative AI in cybersecurity and privacy. *IEEE Access* **11**, 80218–80245 (2023). <https://doi.org/10.1109/ACCESS.2023.3300381>
- [23] 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), <https://openreview.net/forum?id=HcfdQZFZV>
- [24] 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), <https://openreview.net/forum?id=JEflV4nRIH>
- [25] 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. Proceedings of Machine Learning Research*, vol. 202, pp. 22631–22648. PMLR (23–29 Jul 2023), <https://proceedings.mlr.press/v202/longpre23a.html>

- [26] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proceedings of the 2019 International Conference on Learning Representations (2019), <https://openreview.net/forum?id=Bkg6RiCqY7>
- [27] Meta AI: Llama 2 70B Model (2024), <https://huggingface.co/meta-llama/Llama-2-70b>, accessed: 2024-10-27
- [28] Meta AI: Llama 3 8B Model (2024), <https://huggingface.co/meta-llama/Meta-Llama-3-8B>, accessed: 2024-10-27
- [29] Meta AI: Llama 3.1 8B Model (2024), <https://huggingface.co/meta-llama/Llama-3.1-8B>, accessed: 2024-10-20
- [30] Microsoft Research: Phi 3 Mini Instruct 3.8B Model (2024), <https://huggingface.co/microsoft/Phi-3.5-mini-instruct>, accessed: 2024-10-27
- [31] Mistral AI: Mistral 7B Model (2024), <https://huggingface.co/mistralai/Mistral-7B-v0.3>, accessed: 2024-10-27
- [32] 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
- [33] 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)
- [34] Papers with Code: Massive Multitask Language Understanding (MMLU) Benchmark (2024), <https://paperswithcode.com/sota/multi-task-language-understanding-on-mmlu>, accessed: 2024-10-27
- [35] 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), <https://openreview.net/forum?id=GZnsqBwHAG>
- [36] 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), <https://openreview.net/forum?id=hTEGyKf0dZ>
- [37] 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). <https://doi.org/10.1109/ACCESS.2024.3365742>
- [38] robomotic: LLM Guardrails Frameworks (2025), <https://github.com/robomotic/awesome-guide-ai-safety/blob/master/TOOLS.md>, accessed: 2025-04-30
- [39] Roy, S.S., Thota, P., Naragam, K.V., Nilizadeh, S.: From chatbots to phish-bots?: Phishing scam generation in commercial large language models. In: Proceedings of the 2024 IEEE Symposium on Security and Privacy. pp. 36–54 (2024). <https://doi.org/10.1109/SP54263.2024.00182>
- [40] 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). <https://doi.org/10.1109/ACCESS.2024.3403858>

- [41] 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), <https://openreview.net/forum?id=g9diuvxN6D>
- [42] 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. <https://crfm.stanford.edu/2023/03/13/alpaca.html> **3**(6), 7 (2023)
- [43] 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)
- [44] 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). <https://doi.org/10.18653/v1/2020.emnlp-demos.6>
- [45] 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). <https://doi.org/10.1016/j.hcc.2024.100211>
- [46] Zhu, M., Yang, L., Wei, Y., Zhang, N., Zhang, Y.: Locking down the fine-tuned llms safety (2024), <https://arxiv.org/abs/2410.10343>