CyberLLMInstruct: A New Dataset for Analysing Safety of Fine-Tuned LLMs Using Cyber Security Data

Adel ElZemity
University of Kent
Canterbury, United Kingdom

Budi Arief
University of Kent
Canterbury, United Kingdom

Shujun Li
University of Kent
Canterbury, United Kingdom

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. To address these challenges, we developed CyberLLMInstruct, a dataset of 54,928 instruction-response pairs spanning cyber security tasks such as malware analysis, phishing simulations, and zero-day vulnerabilities. The dataset was constructed through a multi-stage process. This involved sourcing data from multiple resources, filtering and structuring it into instruction-response pairs, and aligning it with real-world scenarios to enhance its applicability. Seven open-source LLMs were chosen to test the usefulness of CyberLLMInstruct: 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. In our primary example, we rigorously assess the *safety* of fine-tuned models using the OWASP top 10 framework, finding that fine-tuning reduces safety resilience across *all* tested LLMs and *every* adversarial attack (e.g. the security score of Llama 3.1 8B against prompt injection drops from 0.95 to 0.15). In our second example, we show that these same fine-tuned models can also achieve up to 92.50% accuracy on the CyberMetric benchmark. These findings highlight a trade-off between performance and safety, showing the importance of adversarial testing and further research into fine-tuning methodologies that can mitigate safety risks while still improving performance across diverse datasets and domains. The dataset creation pipeline, along with comprehensive documentation, examples, and resources for reproducing our results, is publicly available at https://github.com/Adelsamir01/CyberLLMInstruct.

ACM Reference Format:
Adel ElZemity, Budi Arief, and Shujun Li. 2025. CyberLLMInstruct: A New Dataset for Analysing Safety of Fine-Tuned LLMs Using Cyber Security Data. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

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 [35] and question answering [33], signalling their transformative potential across various tasks. However, this promise is accompanied by substantial safety risks, requiring focused attention from researchers and practitioners alike [7, 9, 43].

A crucial factor in the success and security of LLMs is the quality of their training data. Well-structured datasets enable models to generalise effectively across diverse cyber security scenarios, yet obtaining such datasets remains a significant challenge. Existing cyber security datasets can be broadly categorised into those designed for traditional machine learning, and those tailored for LLMs. Traditional datasets often fall short in size and diversity, limiting their capacity to support the intricate needs of LLMs, while many LLM-specific datasets remain inaccessible, hindering collaborative research and reproducibility. Furthermore, although realism is critical for accurately reflecting the complexities of real-world cyber security applications, counter-factual and synthetic data are equally important for preparing LLMs to address unknown and emerging threats [13, 41]. Therefore, addressing these dataset limitations is crucial for ensuring that LLMs can reliably detect, prevent, and adapt to evolving cyber threats without being constrained by narrow training data or inaccessible resources.

These limitations highlight the urgent need for more robust, diverse, and realistic datasets to fine-tune LLMs for cyber security applications. This urgency is highlighted by the increasing misuse of generative AI tools like FraudGPT [11] and WormGPT [14] in cyberattacks, enabling adversaries to execute more sophisticated and scalable attacks. A recent study by Falade [11] revealed how malicious LLMs can be exploited to generate phishing lures, impersonation schemes and deepfakes, amplifying the arsenal of cybercriminals and exposing significant vulnerabilities. Another example is the work done by Alotaibi et al. [2], in which techniques like prompt engineering have coerced LLMs into generating malicious content, such as phishing emails and malware code. Other research [34] has shown that vanilla ChatGPT, without requiring advanced adversarial exploits, can create phishing websites that emulate trusted organisations and evade detection by anti-phishing systems. These findings demonstrate the critical need for datasets that prepare LLMs to handle both known and emerging threats in an evolving threat landscape.

Such datasets must encompass a mix of real-world attack data, security logs, and vulnerability reports alongside synthetic and counter-factual data to address known and unknown threats. The inclusion of synthetic data is essential for preparing LLMs to handle emerging attack strategies and novel scenarios. Robustness in this context refers to a dataset's ability to provide diverse, comprehensive, and high-quality data that enhances an LLM's ability

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '25, July 13–18, 2025, Padua, Italy
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

to generalise across a range of cyber security tasks and adversarial conditions. Similarly, the robustness of an LLM relates to its ability to maintain reliable performance under varying conditions, including adversarial attacks, data gaps, and unknown threats.

This paper introduces **CyberLLMInstruct**, a novel dataset designed to enhance LLMs' cyber security capabilities via fine-tuning. In addition to evaluating model behaviours before and after fine-tuning and analysing the safety risks associated with leveraging potentially misused cyber security data, we provide practical insights and guidelines for improving fine-tuning methods to balance performance and safety. CyberLLMInstruct fills a critical gap in existing datasets, which often lack the size, diversity, and practical relevance needed for robust evaluation and development of LLMs in cyber security contexts.

The primary motivation for CyberLLMInstruct is to help researchers systematically identify vulnerabilities in LLMs, allowing them to strengthen safeguards against malicious exploitation. Open-source LLMs, which anyone can freely access and customise, are especially susceptible to misuse because attackers can download and fine-tune them offline, evading public oversight. In contrast, closed-source models are more difficult to examine due to restricted access, posing a barrier to reproducible security research [12]. By offering a dataset specifically geared towards cyber security tasks, CyberLLMInstruct fills a critical gap, enabling researchers to pinpoint weaknesses, develop robust mitigation strategies, and ensure fine-tuning processes focus on model safety as well as performance gains [20].

Contributions. We make the following contributions in this work:

- We release a dataset, **CyberLLMInstruct**, consisting of 54,928 instruction-response pairs that address cyber security tasks. The dataset was constructed through a systematic process, sourcing from authoritative cyber security resources, publicly available threat reports, and simulated scenarios. It also includes advanced topics such as malicious script generation, zero-day vulnerabilities, and adversarial examples for testing model robustness.
- We demonstrate the usefulness of CyberLLMInstruct via two examples. In the first example, CyberLLMInstruct is used to show how safety of fine-tuned LLMs can be assessed using the OWASP top 10 evaluation framework [31]. In the second example, CyberLLMInstruct is used to fine-tune LLMs for improved capabilities in cyber security tasks using the CyberMetric benchmark [38].
- Our two usage examples of CyberLLMInstruct led to key insights about the trade-offs between safety risks and performance gains via fine-tuning LLMs. For instance, while fine-tuning may enhance an LLM's cyber security task performance significantly, it may also reduce its safety resilience. These insights call for more research to safeguard LLMs for cyber security applications.

Overall, this work establishes a strong foundation for advancing the secure deployment of LLMs in cyber security, while providing researchers with a critical resource to further explore the interplay between model performance and safety.

The rest of this paper is organised as follows. Section 2 provides an overview of related work, highlighting existing cyber security datasets and their limitations. Section 3 describes the CyberLLMInstruct dataset, including its creation process, design choices, and unique features. Section 4 presents two examples of using CyberLLMInstruct. Section 5 discusses the trade-offs between performance improvements and security vulnerabilities, the implications of our findings for real-world applications, and the limitations of our work. Section 6 concludes the paper with a summary of key insights and directions for future research.

2 RELATED WORK

Researchers have created various datasets for training LLMs in the field of cyber security, although these datasets often have limitations such as a narrow domain focus and limited generalisability. For instance, Ameri et al. [3] introduced CyBERT, a dataset focused on identifying cyber security feature claims in industrial control systems (ICS) device documents. Similarly, SecQA is a specialised dataset created to evaluate LLMs' understanding of cyber security concepts [23]. CyberMetric offers a comprehensive benchmark dataset containing 10,000 cyber security-related questions spanning nine different cyber security domains [38]. While these datasets serve their intended purposes, their narrow focus can limit the generalisability of LLMs fine-tuned on them.

Several datasets target the application of LLMs to security tasks related to software source code. SVEN is one example, built from a curated selection of existing vulnerability datasets to train LLMs for generating secure code [17]. The authors acknowledged that SVEN does not capture certain security behaviours and programming languages and suggested creating a more comprehensive training dataset to address these limitations. Zhang et al. introduced HackMentor, a cyber security-specific fine-tuned LLM trained on a dataset of 14,000 instructions and 30,000 conversations generated using domain-specific categorisation and expert-curated prompts [44]. Despite its significant performance improvement over baseline models, the authors acknowledged limitations in the dataset's completeness and diversity due to the fragmented and sensitive nature of cyber security data. Jang et al. [22] used a combination of Twitter, blogs, research papers and CVEs to create a dataset for a cyber security-focused BERT-like LLM, which is trained with non-linguistic element aware pre-training method tuned called CyBERTuned. The authors noted that this dataset is limited in its focus on specific non-linguistic elements. Bayer et al. [4] used a diverse corpus of scientific papers, X (formerly Twitter) data, web pages, and the National Vulnerability Database [30] to train a cyber security domain-adapted version of the BERT model called CySecBERT. All of these researchers emphasise the importance of a general cyber security model that can serve as a basis for a variety of tasks.

3 RESOURCE DESCRIPTION

The CyberLLMInstruct dataset, comprising 54,928 records, is designed to cover diverse cyber security topics, formatted in a two-column structure containing instructions and responses. It encapsulates a broad spectrum of cyber security knowledge, including open-source intelligence, threat intelligence, attack techniques, and malware analysis. These categories were chosen based on their foundational importance in prior datasets (e.g., CyBERTuned [22], CySecBERT [4]) and their relevance to real-world applications.

Table 1: Comparison of CyberLLMInstruct with other cyber security datasets.

Dataset	Scope	Malicious Content	Instruction Format	Size	Security Testing	Primary Use
CyBERTuned [22]	Large corpus for pretraining	No	No (text corpus)	~700MB	No direct vulnerability eval	Pretraining LLMs for security awareness
CySecBERT [4]	Security news, CVE reports	No	No (text corpus)	~4.3M documents	Limited	Domain-adaptive BERT for security tasks
SecQA [23]	Multiple-choice Q&A	No	No (Q&A pairs)	127 Qs (v1) 115 Qs (v2)	Not evaluated	Basic security knowledge benchmarking
CyberMetric [38]	Large cyber security Q&A benchmark	No	No (Q&A format)	10,000 questions	Minimal	Evaluating LLM knowledge in cybersecurity
SVEN [17]	Secure vs. insecure code pairs	Insecure code snippets	No (code diffs)	803 fix pairs	Some (prefix-tuning for safe vs. unsafe code)	Code generation control (secure/insecure outputs)
CyberLLMInstruct	Instruction-response cyber security dataset	**Yes** (malicious + benign)	**Yes** (full instruction format)	54,928 records	**Yes**, tested with OWASP framework	Fine-tuning LLMs, adversarial testing, security training

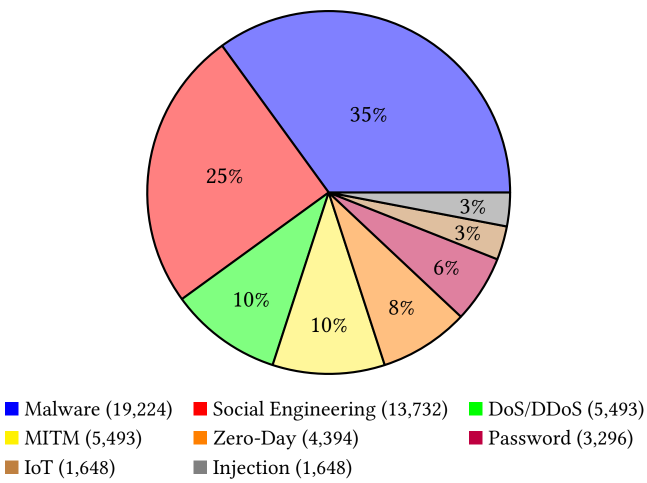

Figure 1: Security categories in CyberLLMInstruct dataset.

Prompts guide LLMs through advanced scenarios, e.g., designing phishing campaigns, countering cyber security breaches, conducting social engineering simulations, exploring IoT security, network reconnaissance, and applications of such techniques. Unlike datasets that focus solely on defensive or descriptive cyber security knowledge, CyberLLMInstruct intentionally includes directly malicious content alongside benign discussions obtained mostly from educational resources such as Capture the Flag (CTF) challenges. This includes offensive security techniques, adversarial prompts, and examples of harmful code segments. For example, records related to malware do not merely describe malware threats but also include real-world attack simulations, malware demonstrations, and evasion techniques. CyberLLMInstruct dataset is designed for researchers and practitioners with intermediate to advanced knowledge in cyber security, particularly those working on LLM fine-tuning, adversarial testing, and cyber threat analysis. See Table 1 for a concise comparison of CyberLLMInstruct with related datasets.

The CyberLLMInstruct security categories were assigned based on the IBM X-Force Threat Intelligence Index 2024 [19], ensuring their relevance to the evolving threat landscape in cyber security. The selected categories address gaps identified in prior datasets and include cutting-edge topics critical to modern cyber security challenges. Practical examples – such as those derived from real-world cyber incidents – cover significant areas, including malware, social engineering, zero-day exploits, and IoT vulnerabilities.

Malware, observed in 43% of incidents globally according to the IBM X-Force Threat Intelligence Index 2024, plays an important role as a dominant attack method. In our dataset, malware accounts for **35%** of the records, reflecting its prevalence across various subcategories, including ransomware, Trojans, spyware, and worms. This allocation ensures the dataset emphasises malware's significant role in unauthorised access, data theft, and system disruption, which aligns with its prominence in real-world cyber threats.

Social engineering and phishing, representing **25%** of the dataset, leverage human manipulation to compromise sensitive information. Denial-of-service (DoS, including distributed DoS – DDoS) attacks, and man-in-the-middle (MITM) attacks each account for **10%** of the records, focusing on system disruption and eavesdropping, respectively. Zero-day exploits (**8%**) highlight the risks associated with unpatched vulnerabilities, while password attacks (**6%**) target weak or stolen credentials. Emerging threats such as Internet of Things (IoT) attacks (**3%**) and injection attacks (**3%**) showcase the vulnerabilities of interconnected devices and database systems.

Categories were assigned to the records using a semi-automatic methodology, combining manual (by the first author of the paper) and automated approaches utilising ChatGPT-4.0 and Gemini 1.5 Flash to ensure precision while optimising the efficiency of the process. The distribution of categories is detailed in Figure 1, showing how records were allocated in the dataset to align with the significance of each category in modern cyber security. For more details on how the categories were selected, please refer to Section 3.2.

3.1 Dataset Creation Process

As shown in Figure 2, the creation of the CyberLLMInstruct dataset followed a planned and multi-stage process that aimed at ensuring the inclusion of diverse and relevant cyber security knowledge and skills. The primary objective was to develop a dataset enabling the fine-tuning of LLMs for various practical applications in cyber security. This dataset equips LLMs with the necessary fine-tuning data to enhance their comprehension of cyber security tasks, supporting practical applications across diverse security domains. Given the

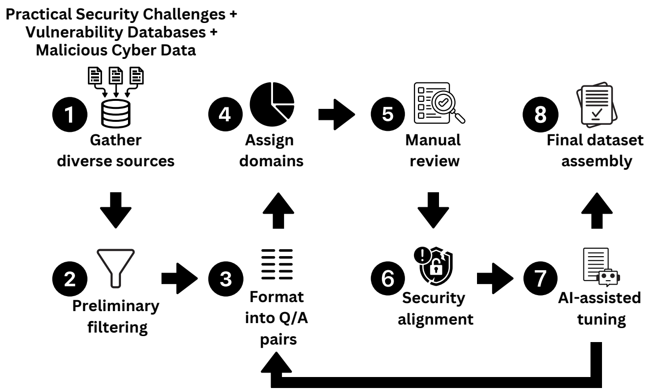

Figure 2: A high-level overview of the CyberLLMInstruct dataset creation process, illustrating the steps from gathering diverse sources (including malicious cyber data) through filtering, formatting into Q/A pairs, domain assignment, manual review, security alignment, AI-assisted tuning, and final dataset assembly.

increasing reliance on LLMs in cyber security [21], the research community using this dataset is likely to grow in the coming years as the demand for secure and specialised AI applications expands.

The process began with Stage 1 “Data Collection”, where raw content was gathered from different sources covering real-world data sources. Authoritative resources, including NIST standards, research papers published in reputable venues, and industry reports (e.g., from the SANS Institute), were prioritised. Real-world incident reports, including data from the Common Vulnerabilities and Exposures (CVE) database, were also incorporated, along with hands-on examples from CTF challenges. Additionally, malicious cyber data, such as phishing campaigns, malware code, and known exploits, were included in the dataset to enhance the LLM's effectiveness when confronted with malicious prompts. The full list of these resources and their justifications is provided with the dataset.

In Stage 2 “Preliminary Filtering”, irrelevant materials for cyber security were removed after manual inspection by the first author.

During Stage 3 “Data Structuring”, the dataset was formatted into a consistent two-column instruction-response format, transforming raw content into prompts and corresponding responses.

In Stage 4 “Domain Assignment”, data were categorised semi-automatically, distributing records across eight categories to comprehensively cover a broad range of cyber security topics. These categories included “Malware”, “Social Engineering and Phishing”, “DoS/DDoS Attacks”, “MITM Attacks”, “Zero-Day Exploits”, “Password Attacks”, “IoT Attacks”, and “Injection Attacks”, as outlined in Figure 1. This categorisation ensures that the dataset encompasses diverse and critical areas within the cyber security domain.

To maintain high quality, manual inspection was primarily conducted during Stage 5 “Manual Review”, where the paper's first author verified the technical accuracy, relevance, and educational value of entries. Both benign and malicious entries were scrutinised to eliminate trivial or misleading content, ensuring alignment with current threats and best practices. A random sampling approach was used, with efforts made to balance the manual review across categories. Categories with higher percentages of entries received proportionally more attention to ensure comprehensive coverage. This selective yet systematic approach aimed to maximise reliability while managing the scope of the review, though some unreviewed records may carry a marginal risk of reduced accuracy.

In Stage 6 “Security Alignment”, adversarial considerations and red-teaming input were integrated to ensure that malicious examples were designed to realistically test the limits of the LLM's compliance mechanisms. The focus was on exposing the LLM to adversarial and malicious prompts, alongside benign cyber security data, to challenge its ability to identify and respond to high-risk queries responsibly. This dataset was intentionally designed to push the boundaries of the LLM's safety guardrails, recognising that in such scenarios, the LLM might fail or exhibit vulnerabilities, providing critical insights into its limitations.

Next, Stage 7 “AI-Assisted Tuning” used advanced LLMs (Gemini 1.5 Pro and GPT-4) to enhance language clarity, detail, and consistency. This refinement focused on improving the quality of both prompts and answers, making them more comprehensive and detailed. The process also addressed errors, reduced biases, and improved coherence across both malicious and benign examples. Following this stage, a final round of manual review was conducted for a random selected subset to ensure the outputs met the desired standards of precision, accuracy, and completeness.

Finally, Stage 8 “Final Integration and Packaging” consolidated all reviewed, categorised, and security-aligned records into a cohesive dataset. The result is a comprehensive resource designed to facilitate rigorous testing of LLMs, to enhance LLMs' performance on cyber security tasks, and to challenge them with adversarial content.

3.2 Design Choices

The design choices made in constructing the CyberLLMInstruct dataset were guided by the need to create a resource that would effectively facilitate fine-tuning of LLMs for practical cyber security applications. The justifications for the three key design choices are provided below.

(1) The choice to use an instruction-response format is supported by several sources [18, 42, 43, 45]. They highlight instruction tuning's effectiveness for enhancing LLM performance. Yang et al. [42] discussed how instruction-triggered attacks exploit fine-tuning through poisoned instructions. Zhao et al. [45] explained how instruction tuning helps align LLMs with human values. This format was chosen to provide LLMs with clear guidance on cyber security tasks. Instructions simulate real-world scenarios, prompting the LLM to generate appropriate responses, such as code, vulnerability analysis, or mitigation strategies.
(2) The decision to include both foundational concepts and advanced hands-on scenarios was guided by the objective of creating a comprehensive dataset. Ferrag et al. [13] and Tann et al. [37] emphasised the importance of varying question complexity to assess LLMs' understanding. Tihanyi et al. [38] highlighted the need to include challenging questions to push LLMs' knowledge limits. The CyberLLMInstruct dataset contains detailed instructions and comprehensive responses, including code snippets, step-by-step procedures,

and in-depth explanations. Such a level of details facilitates practice-oriented learning, allowing LLMs to learn by examples.

(3) The distribution of categories in CyberLLMInstruct aligns with both their real-world prevalence and the availability of structured data, as previously discussed in Section 3. Higher representation of categories such as "Malware" (35%) and "Social Engineering and Phishing" (25%) reflects their frequent occurrence in cyber incidents and extensive documentation. Conversely, categories like "Zero-Day Exploits" (8%), "Password Attacks" (6%), "IoT Attacks" (3%), and "Injection Attacks" (3%) are less represented due to the challenges in sourcing publicly available structured data. This distribution mirrors the natural imbalance found in cyber security threat intelligence [36], as detailed earlier.

4 DATASET UTILITY

In this section, we present two application examples of CyberLLMInstruct. First, in Section 4.2, we examine the safety risks introduced by fine-tuning LLMs on this dataset, leveraging the OWASP top 10 framework. Then, in Section 4.3, we explore whether such fine-tuning also improves performance on cyber security tasks. To achieve this, we start by explaining the fine-tuning methodology, including the computational setup, model selection, and training configuration. The evaluation is then divided into two parts: (1) Safety Analysis, where we assess vulnerabilities introduced by fine-tuning, and (2) Performance Assessment, where we measure improvements using the CyberMetric benchmark.

4.1 Fine-Tuning Process

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. Each model was fine-tuned within a period of less than two days, ensuring that the hardware was utilised efficiently without excessive resource consumption.

The models selected for fine-tuning were Phi 3 Mini 3.8B [28], Mistral 7B [29], Qwen 2.5 7B [1], Llama 3 8B [26], Llama 3.1 8B [27], Gemma 2 9B [15], and Llama 2 70B [25]. These models were chosen due to their strong performance on the Massive Multitask Language Understanding (MMLU) benchmark [32], 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 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

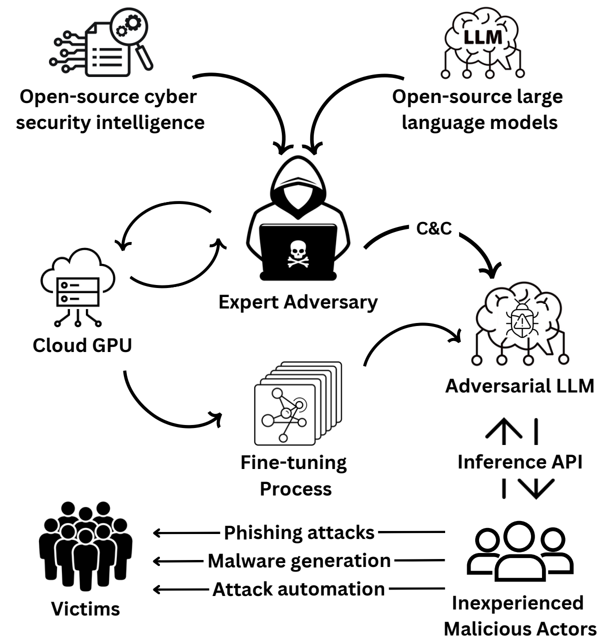

Figure 3: Abstraction of the adversarial LLM threat model.

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, which has been discussed throughout the paper as a core resource. Fine-tuning was conducted using the SFTTrainer from the TRL library [39], with training configured using TrainingArguments from the Transformers library [40]. The configuration included a batch size of 2 per device, with gradient accumulation steps set to 4, facilitating stable training with limited memory. The models were fine-tuned over 10 epochs, with a learning rate of 2×10^{-4} chosen for optimal convergence. Additionally, 16-bit floating point precision was used to optimise memory usage, unless `bfloat16` precision was supported by the hardware. The AdamW optimiser [24] with a weight decay of 0.01 was employed to prevent overfitting, and a linear scheduler was used to control 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.

4.2 Example 1: Safety Analysis

Our first example focuses on model safety. Given the potential for adversarial uses, it is crucial to examine whether any performance gains come at the expense of resilience against malicious attacks. We use the OWASP top 10 framework [31] to assess how fine-tuning affects each LLM's susceptibility to various vulnerabilities.

4.2.1 Threat Model. Although CyberLLMInstruct is designed to enhance LLMs in legitimate cyber security tasks, its misuse poses significant risks. As shown in Figure 3, 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 3 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.

4.2.2 OWASP top 10 Evaluation Setup. In this first example, we use the OWASP top 10 framework [31] to investigate whether the performance improvements (that can be seen later in Example 2 in Section 4.3) come at the expense of safety. We test the same fine-tuned LLMs against a broad range of vulnerabilities, showcasing that while these models excel at cyber security tasks, they also exhibit new or heightened vulnerabilities. OWASP top 10 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. The 2025 edition of the OWASP top 10 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 the DeepEval framework [8], 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 the DeepEval framework. 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. [16], 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. [10], 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.

4.2.3 Results. Table 2 presents a comprehensive analysis of how base and fine-tuned LLMs perform against OWASP top 10 vulnerabilities in cyber security applications. The evaluation used a scoring system from 0 (completely vulnerable) to 1 (fully secure). Figure 4

Table 2: Performance of base (green) and fine-tuned (red) LLMs against OWASP Top 10 vulnerabilities (scores range from 0, representing completely vulnerable, to 1, fully secure).

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	0.80 / 0.25	0.85 / 0.20
Sensitive Info. Disclosure	0.89 / 0.45	0.85 / 0.30	0.86 / 0.35	0.84 / 0.40	0.90 / 0.25	0.78 / 0.30	0.82 / 0.42
Supply Chain	0.87 / 0.48	0.82 / 0.40	0.85 / 0.45	0.86 / 0.50	0.88 / 0.30	0.80 / 0.35	0.84 / 0.32
Data and Model Poisoning	0.85 / 0.40	0.87 / 0.25	0.89 / 0.30	0.88 / 0.32	0.93 / 0.20	0.84 / 0.25	0.90 / 0.22
Improper Output Handling	0.93 / 0.48	0.89 / 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	0.89 / 0.28
Prompt Leakage	0.85 / 0.33	0.85 / 0.25	0.89 / 0.30	0.84 / 0.35	0.91 / 0.20	0.84 / 0.35	0.86 / 0.22
Embedding Weaknesses	0.89 / 0.42	0.90 / 0.35	0.91 / 0.40	0.82 / 0.35	0.93 / 0.30	0.91 / 0.32	0.88 / 0.42
Misinformation	0.93 / 0.38	0.87 / 0.30	0.89 / 0.35	0.84 / 0.30	0.95 / 0.25	0.91 / 0.40	0.90 / 0.58
Unbounded Consumption	0.94 / 0.48	0.90 / 0.45	0.91 / 0.48	0.88 / 0.40	0.94 / 0.40	0.93 / 0.50	0.92 / 0.42

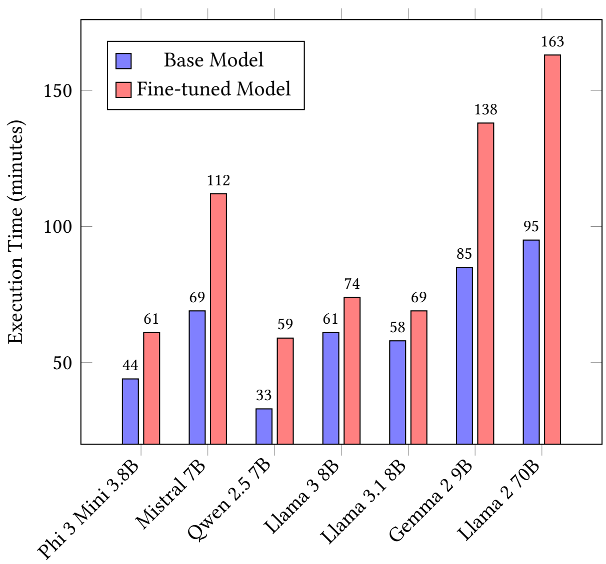

Figure 4: Execution times for base and fine-tuned LLMs (ordered from smallest to largest model).

complements this by showing the execution 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.

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 security across all vulnerability categories. Input manipulation vulnerabilities (particularly "Prompt Injection") and data exposure risks ("Sensitive Information Disclosure") emerged as the most 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.

Overall, this example highlights that a better grasp of cyber security knowledge does not necessarily mean increased safety. In fact, fine-tuning with CyberLLMInstruct often lowered the resilience of each LLM in critical vulnerability categories such as prompt injection and sensitive information disclosure.

4.3 Example 2: Performance Assessment

In our second example, we use the CyberMetric benchmark [38] to demonstrate how CyberLLMInstruct can also improve LLM performance on cyber security tasks. CyberMetric is a domain-specific

Table 3: Accuracy results (%) for different base (before arrow) and fine-tuned (after arrow) LLMs on the CyberMetric benchmark.

LLM Model	80 Q	500 Q	2k Q	10k Q
Phi 3 Mini 3.8B	5.00 → 53.75	5.00 → 40.60	4.41 → 28.75	4.80 → 19.18
Mistral 7B	78.75 → 81.94	78.40 → **91.80**	76.40 → 91.10	74.82 → 88.89
Qwen 2.5 7B	43.75 → 73.75	58.00 → 64.60	55.75 → 69.00	54.09 → 66.10
Llama 3 8B	38.75 → 82.50	35.80 → 48.00	37.00 → 49.45	36.00 → 50.75
Llama 3.1 8B	81.25 → **92.50**	76.20 → 87.80	73.05 → **91.25**	71.25 → **88.50**
Gemma 2 9B	42.50 → 78.75	37.20 → 52.80	36.00 → 50.44	43.28 → 59.79
Llama 2 70B	75.00 → 90.00	73.40 → 78.40	71.60 → 84.00	66.10 → 74.82

benchmark with expert-validated questions. By fine-tuning LLMs on CyberLLMInstruct, we show noticeable accuracy gains, indicating that LLMs acquire more robust cyber security knowledge.

This evaluation focused on measuring the accuracy of LLMs across various question sets, with results validated using parsing scripts and a secondary judge LLM to extract answers robustly. A series of iterative experiments were conducted to evaluate and improve the accuracy of Llama 3 8B, which was used for calibration and ensuring we have a robust answer extraction method on the CyberMetric 80-question dataset.

The answers for the 80-question dataset were manually verified by the first author of the paper for correctness, ensuring the reliability of the results. For the 500, 2k, and 10k sets of questions, a random sampling method of 80 questions was used to substantiate the claim that the results generated by the judge LLM are correct. The initial runs used basic parsing to extract answers, resulting in limited accuracy. Subsequent improvements included improved parsing rules and the use of a judge LLM (ChatGPT 4o and Gemini 1.5 Flash) to extract answers, which led to perfecting answer extraction. The choice of these models was based on their superior performance shown in the MMLU benchmark.

4.3.1 Results. The iterative evaluation of Llama 3 8B demonstrated significant improvements in accuracy, reaching a peak of 82.50% after introducing an optimised judge LLM. The baseline accuracy of 81.25% was evaluated and confirmed to match the results reported in [38]. These results highlight the importance of robust answer extraction methodologies in fine-tuned LLM evaluation. Fine-tuning on the CyberLLMInstruct dataset consistently improved the performance of all evaluated LLMs, as shown in Table 3.

The best post-fine-tuning accuracy was achieved by Llama 3.1 8B, which attained 92.50% on the 80-question dataset, demonstrating the highest level of cyber security-specific expertise. On average, fine-tuning led to significant accuracy improvements across all models, with an average improvement of 26.88% on the 80-question dataset, 14.29% on the 500-question dataset, 15.68% on the 2000-question dataset, and 13.96% on the 10,000-question dataset. The median improvements were similarly substantial, with a 30% increase for the 80-question dataset and over 12.20% for larger datasets.

These results highlight the efficacy of fine-tuning in enhancing both the overall and detailed cyber security knowledge of LLMs, even across varying dataset sizes and complexities.

It is important to note that, while fine-tuning introduces new vulnerabilities, domain experts with strong cyber security knowledge may still benefit greatly from fine-tuned LLMs. Their expertise can help mitigate risks introduced by malicious use or erroneous outputs, thereby allowing them to leverage the improved task-specific performance without fully compromising on safety.

5 FURTHER DISCUSSIONS

This section explores the key insights and implications drawn from the two examples in Section 4. While our findings first highlight critical vulnerabilities and limitations introduced by fine-tuning, they also reveal notable advancements in domain-specific performance. We focus on three main aspects: dataset insights from example use cases, differences in vulnerability profiles across model architectures, and the limitations of current benchmarking frameworks.

5.1 Dataset Insights From Example Use Cases

The experimental results reveal that the CyberLLMInstruct dataset's broad coverage of adversarial prompts – ranging from social engineering lures to code obfuscation – exposed nuanced weaknesses in fine-tuned models (see Table 2 for a summary). Realistic malicious samples, although implemented in benign pseudocode form, provided essential stress tests that brought to light how specialised security data can erode the baseline safety. In particular, the dataset's multi-category construction (e.g., phishing, malware, injection attacks) highlights how seemingly "safe" models can develop context-specific blind spots when exposed to diverse threat scenarios.

The fact that CyberLLMInstruct includes both benign and malicious samples may imply that the introduced vulnerabilities by fine-tuning may arise not only from the malicious information included in the fine-tuning dataset, but also from some intrinsic weakness of the model itself that "over-react" to the malicious information to fail any safety mechanisms. How benign and malicious samples in fine-tuning datasets like CyberLLMInstruct can affect the balance between a fine-tuned LLM's performance and safety resilience remains an interesting research question. Our experimental results from Example 1 revealed some initial insights, but cannot give conclusive answers to this question due to its limited scope. We therefore call for more research on this topic, which can be supported by CyberLLMInstruct and its extended versions.

5.2 Model-Specific Vulnerability Profiles

Our results suggest that model size and architecture affect safety resilience following fine-tuning on the CyberLLMInstruct dataset, although the effect varies across attack categories. Preliminary evidence (see Figure 5) indicates that, while smaller models, such as Phi

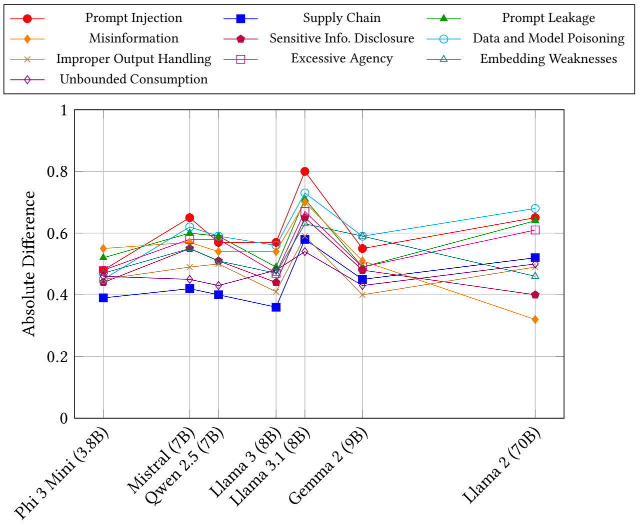

Figure 5: Absolute difference before and after fine-tuning OWASP top 10 vulnerabilities for all tested LLMs of varying sizes. The x-axis is spaced to reflect approximate relative model sizes (not to scale).

3 Mini (3.8B), maintain relatively higher safety scores in some categories, larger models, such as Llama 2 (70B), exhibit greater safety degradation. Notably, the relationship is not strictly monotonic; Llama 3.1 (8B) shows the most significant vulnerability, suggesting that architectural choices and fine-tuning methodologies may also play a crucial role.

We can also observe that vulnerability patterns depend on the type of attack. Some models remain stable in areas like "Improper Output Handling", while others, such as Llama 3.1 (8B), show substantial declines in "Prompt Injection" and "Sensitive Information Disclosure". These findings highlight the need for further research with a larger sample and additional benchmarks to better understand model-specific safety risks and mitigation strategies.

5.3 Limitations

While the OWASP top 10 and CyberMetric evaluations offer valuable quantitative insights into safety and performance, these benchmarks operate under controlled conditions and thus may not fully capture the complexities of real-world applications. For instance, the static OWASP evaluations do not account for adaptive adversaries or evolving threats and do not measure how localised vulnerabilities could cascade through interconnected systems. Similarly, in sectors such as healthcare or power grids, incomplete or noisy data might significantly degrade the model's performance – such scenarios are not currently reflected in the CyberMetric's curated questions.

Both the judge LLM and DeepEval tools used in our tests can introduce biases or fail to represent model behaviours across domain-specific edge cases. The CyberLLMInstruct dataset itself is not without challenges, including potential biases stemming from its data sources and an imbalanced distribution of malicious content versus benign samples. 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.

An additional limitation is the predominantly single-step nature of our evaluations, which does not fully account for multi-turn, agent-like interactions often encountered in real-world deployments. Adversarial prompts may evolve across multiple conversation steps, potentially revealing deeper or more covert vulnerabilities. Future work should incorporate multi-step or chain-of-thought scenarios to investigate how these models behave under iterative adversarial pressures.

Addressing these gaps will likely involve more dynamic and domain-specific testing frameworks, including real-time adversarial feedback and cross-functional testbeds capable of measuring cascading impacts. By refining benchmarking methods, balancing and validating CyberLLMInstruct dataset, and systematically examining model architectures, we can develop more robust, trustworthy LLMs suitable for deployment in complex, high-stakes environments.

6 CONCLUSION

This paper introduced CyberLLMInstruct, a dataset specifically designed to evaluate the safety risks of fine-tuning LLMs on cyber security data. In our primary example, we demonstrated that fine-tuned models exhibit increased vulnerabilities, including prompt injection and sensitive information disclosure, as identified using the OWASP top 10 framework. These findings highlight a critical trade-off, as fine-tuning enhances cyber security-specific knowledge while simultaneously reducing safety resilience. In our second example, we explored whether fine-tuning also improves performance, finding that models achieve up to 92.50% accuracy on the CyberMetric benchmark. These results emphasise the importance of fine-tuning methodologies that prioritise safety while maintaining performance benefits. Future research should focus on mitigating these vulnerabilities while ensuring LLMs remain effective for cyber security applications.

There is a pressing need for in-depth investigations into the harm caused by malicious LLMs circulating on the dark web. These models have been shown to automate phishing campaigns, malware generation, and social engineering attacks. Analysing their effectiveness and impact will help design robust countermeasures and inform guidelines to prevent the misuse of open-source LLMs. These efforts will advance the secure deployment of LLMs and enhance their resilience in critical applications.

REFERENCES

[1] Alibaba DAMO Academy. 2024. *Qwen 2.5 Coder 7B Model.* https://huggingface.co/Qwen/Qwen2.5-Coder-7B Accessed: 2024-10-27.
[2] Lara Alotaibi, Sumayyah Seher, and Nazeeruddin Mohammad. 2024. 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* (2024), 1304–1311. https://doi.org/10.1109/ICETSIS61505.2024.10459698
[3] Kimia Ameri, Michael Hempel, Hamid Sharif, et al. 2021. CyBERT: Cybersecurity Claim Classification by Fine-Tuning the BERT Language Model. *Journal of Cybersecurity and Privacy* 1, 4 (2021), 615–637. https://doi.org/10.3390/jcp1040031
[4] Markus Bayer, Philipp Kuehn, Ramin Shanehsaz, and Christian Reuter. 2024. CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain. *ACM Transactions on Privacy and Security* 27, 2, Article 18 (2024), 20 pages. https://doi.org/10.1145/3652594
[5] Stefano Caneppele and Amandine da Silva. 2022. Cybercrime. In *Research Handbook of Comparative Criminal Justice*. Edward Elgar Publishing, 243–260.

[6] Helena Carrapico and Benjamin Farrand. 2017. 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*. Routledge, 146–156. https://www.taylorfrancis.com/chapters/edit/10.4324/9781315645629-12/cyber-crime-fragmented-policy-field-context-area-freedom-security-justice-helena-carrapico-benjamin-farrand
[7] P. V. Sai Charan, Hrushikesh Chunduri, P. Mohan Anand, and Sandeep K. Shukla. 2023. From Text to MITRE Techniques: Exploring the Malicious Use of Large Language Models for Generating Cyber Attack Payloads. arXiv:2305.15336 [cs.CR]
[8] Confident AI. 2024. *DeepEval: The Open-Source LLM Evaluation Framework*. https://docs.confident-ai.com/docs/red-teaming-introduction
[9] Erik Derner, Kristina Batistic, Jan Zahálka, and R. Babuška. 2023. A Security Risk Taxonomy for Prompt-Based Interaction With Large Language Models. *IEEE Access* 12 (2023), 126176–126187. https://doi.org/10.1109/ACCESS.2024.3450388
[10] Kohei Dozono, T. Gasiba, and Andrea Stocco. 2024. Large Language Models for Secure Code Assessment: A Multi-Language Empirical Study. arXiv:2408.06428 [cs.SE]
[11] Polra Victor Falade. 2023. Decoding the Threat Landscape: ChatGPT, FraudGPT, and WormGPT in Social Engineering Attacks. arXiv:2310.05595 [cs.CR]
[12] Mohamed Amine Ferrag, Fatima Alwahedi, Ammar Battah, Bilel Cherif, Abdechakour Mechri, Norbert Tihanyi, Tamas Bisztray, and Merouane Debbah. 2024. Generative AI in Cybersecurity: A Comprehensive Review of LLM Applications and Vulnerabilities. arXiv:2405.12750 [cs.CR]
[13] Mohamed Amine Ferrag, Mthandazo Ndhlovu, Norbert Tihanyi, Lucas C. Cordeiro, Merouane Debbah, Thierry Lestable, and Narinderjit Singh Thandi. 2024. Revolutionizing Cyber Threat Detection With Large Language Models: A Privacy-Preserving BERT-Based Lightweight Model for IoT/IIoT Devices. *IEEE Access* 12 (2024), 23733–23750. https://doi.org/10.1109/ACCESS.2024.3363469
[14] Mohamed Fazil Mohamed Firdhous, Walid Elbreiki, Ibrahim Abdullahi, B. H. Sudantha, and Rahmat Budiarto. 2023. WormGPT: A Large Language Model Chatbot for Criminals. In *Proceedings of the 2023 24th International Arab Conference on Information Technology*. https://doi.org/10.1109/ACIT58888.2023.10453752
[15] Google AI. 2024. *Gemma 2 9B Model*. https://huggingface.co/google/gemma-2-9b Accessed: 2024-10-27.
[16] Maanak Gupta, Charankumar Akiri, Kshitiz Aryal, Eli Parker, and Lopamudra Praharaj. 2023. From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. *IEEE Access* 11 (2023), 80218–80245. https://doi.org/10.1109/ACCESS.2023.3300381
[17] Jingxuan He and Martin Vechev. 2023. Large Language Models for Code: Security Hardening and Adversarial Testing. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 1865–1879. https://doi.org/10.1145/3576915.3623175
[18] Md Imran Hossen, Jianyi Zhang, Yinzhi Cao, and Xiali Hei. 2024. Assessing Cybersecurity Vulnerabilities in Code Large Language Models. arXiv:2404.18567 [cs.CR]
[19] IBM Corporation. 2024. *IBM X-Force Threat Intelligence Index 2024*. https://www.ibm.com/reports/threat-intelligence Accessed: 2024-09-23.
[20] Ryosuke Ishibashi, K. Miyamoto, Chansu Han, Tao Ban, Takeshi Takahashi, and Jun'ichi Takeuchi. 2022. Generating Labeled Training Datasets Towards Unified Network Intrusion Detection Systems. *IEEE Access* 10 (2022), 53972–53986. https://doi.org/10.1109/ACCESS.2022.3176098
[21] Arun Iyengar and Ashish Kundu. 2023. Large Language Models and Computer Security. In *Proceedings of the 2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications*. 307–313. https://doi.org/10.1109/TPS-ISA58951.2023.00045
[22] Eugene Jang, Jian Cui, Dayeon Yim, Youngjin Jin, Jin-Woo Chung, Seungwon Shin, and Yongjae Lee. 2024. Ignore Me But Don't Replace Me: Utilizing Non-Linguistic Elements for Pretraining on the Cybersecurity Domain. In *Findings of the Association for Computational Linguistics: NAACL 2024*. ACL, 29–42. https://doi.org/10.18653/v1/2024.findings-naacl.3
[23] Zefang Liu. 2023. SecQA: A Concise Question-Answering Dataset for Evaluating Large Language Models in Computer Security. arXiv:2312.15838 [cs.CL]
[24] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of the 2019 International Conference on Learning Representations*. https://openreview.net/forum?id=Bkg6RiCqY7
[25] Meta AI. 2024. *Llama 2 70B Model*. https://huggingface.co/meta-llama/Llama-2-70b Accessed: 2024-10-27.
[26] Meta AI. 2024. *Llama 3 8B Model*. https://huggingface.co/meta-llama/Meta-Llama-3-8B Accessed: 2024-10-27.
[27] Meta AI. 2024. *Llama 3.1 8B Model*. https://huggingface.co/meta-llama/Llama-3.1-8B Accessed: 2024-10-20.
[28] Microsoft Research. 2024. *Phi 3 Mini Instruct 3.8B Model*. https://huggingface.co/microsoft/Phi-3.5-mini-instruct Accessed: 2024-10-27.
[29] Mistral AI. 2024. *Mistral 7B Model*. https://huggingface.co/mistralai/Mistral-7B-v0.3 Accessed: 2024-10-27.
[30] National Institute of Standards and Technology (NIST), USA. 2024. *National Vulnerability Database (NVD)*. https://nvd.nist.gov/ Accessed: 2024-08-04.
[31] OWASP Foundation. 2025. OWASP Top 10 for Large Language Model Applications. https://owasp.org/www-project-top-10-for-large-language-model-applications/ Accessed: 2024-12-16.
[32] Papers with Code. 2024. *Massive Multitask Language Understanding (MMLU) Benchmark*. https://paperswithcode.com/sota/multi-task-language-understanding-on-mmlu Accessed: 2024-10-27.
[33] Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, et al. 2024. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access* 12 (2024), 26839–26874. https://doi.org/10.1109/ACCESS.2024.3365742
[34] Sayak Saha Roy, Poojitha Thota, Krishna Vamsi Naragam, and Shirin Nilizadeh. 2024. From Chatbots to Phishbots?: Phishing Scam Generation in Commercial Large Language Models. In *Proceedings of the 2024 IEEE Symposium on Security and Privacy*. 36–54. https://doi.org/10.1109/SP54263.2024.00182
[35] Zoltán Ságodi, István Siket, and Rudolf Ferenc. 2024. Methodology for Code Synthesis Evaluation of LLMs Presented by a Case Study of ChatGPT and Copilot. *IEEE Access* 12 (2024), 72303–72316. https://doi.org/10.1109/ACCESS.2024.3403858
[36] Strahil A. Sokolov, Teodor B. Iliev, and Ivaylo S. Stoyanov. 2019. Analysis of Cybersecurity Threats in Cloud Applications Using Deep Learning Techniques. In *Proceedings of the 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics*. 441–446. https://doi.org/10.23919/MIPRO.2019.8756755
[37] Wesley Tann, Yuancheng Liu, Jun Heng Sim, Choon Meng Seah, and Ee-Chien Chang. 2023. Using Large Language Models for Cybersecurity Capture-The-Flag Challenges and Certification Questions. arXiv:2308.10443 [cs.AI]
[38] Norbert Tihanyi, Mohamed Amine Ferrag, Ridhi Jain, Tamas Bisztray, and Merouane Debbah. 2024. CyberMetric: A Benchmark Dataset based on Retrieval-Augmented Generation for Evaluating LLMs in Cybersecurity Knowledge. In *Proceedings of the 2024 IEEE International Conference on Cyber Security and Resilience*. 296–302. https://doi.org/10.1109/CSR61664.2024.10679494
[39] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformer Reinforcement Learning. https://github.com/huggingface/trl.
[40] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. ACL, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6
[41] HanXiang Xu, ShenAo Wang, Ningke Li, Yanjie Zhao, Kai Chen, Kailong Wang, Yang Liu, Ting Yu, and HaoYu Wang. 2024. Large Language Models for Cyber Security: A Systematic Literature Review. arXiv:2405.04760 [cs.CR]
[42] Haomiao Yang, Kunlan Xiang, Mengyu Ge, Hongwei Li, Rongxing Lu, and Shui Yu. 2024. A Comprehensive Overview of Backdoor Attacks in Large Language Models Within Communication Networks. *IEEE Network* 38, 6 (2024), 211–218. https://doi.org/10.1109/MNET.2024.3367788
[43] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly. *High-Confidence Computing* 4, 2, Article 100211 (2024), 21 pages. https://doi.org/10.1016/j.hcc.2024.100211
[44] Jie Zhang, Hui Wen, Liting Deng, Mingfeng Xin, Zhi Li, Lun Li, Hongsong Zhu, and Limin Sun. 2023. HackMentor: Fine-Tuning Large Language Models for Cybersecurity. *Proceedings of the 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications* (2023), 452–461. https://doi.org/10.1109/TrustCom60117.2023.00076
[45] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. arXiv:2303.18223 [cs.CL]