The AAAI-24 Workshop on Artificial Intelligence for Cyber Security (AICS)

CyberBench: A Multi-Task Benchmark for Evaluating Large Language Models in Cybersecurity

Zefang Liu[1*], **Jialei Shi**[1*], **John F. Buford**[1]
[1]JPMorgan Chase
Palo Alto, California 94304 USA
zefang.liu@jpmchase.com

Abstract

We present CyberBench and CyberInstruct, two innovative tools designed to enhance the application of large language models (LLMs) in the cybersecurity field. Firstly, CyberBench is a domain-specific multi-task benchmark tailored for assessing LLM performance in cybersecurity-related tasks. As the first benchmark suite for LLMs in cybersecurity, CyberBench fills a crucial gap in the current practice by providing a general and consistent approach and addressing coverage limitations of prior language model evaluations in this domain. We showcase the results of using CyberBench to evaluate more than ten generative LLMs. Secondly, CyberInstruct is a family of generative LLMs produced through instruction-tuning from open LLMs with a cybersecurity corpus. Experimental results of CyberInstruct exhibit comparable performance to large proprietary LLMs in the cybersecurity domain, underscoring the effectiveness of our fine-tuning strategy. Our work contributes to the understanding of LLMs' potential in cybersecurity and establishes a solid foundation for future research and development.

1 Introduction

In the rapidly evolving digital landscape, cybersecurity has emerged as a critical concern for individuals, businesses, and governments. Over the next decade, the importance of cybersecurity is expected to grow exponentially as increasing numbers of devices, systems, and infrastructures become interconnected. However, cybersecurity is a complex domain. From an application perspective, cybersecurity practice ranges from physical layer security to application layer security. Deep system knowledge of operating systems, network protocols, infrastructure, and their associated attack surface and defense components is needed. From a terminology perspective, Wikipedia[1] uses over one thousand related categories and pages to tag its cybersecurity related articles, and National Institute of Standards and Technology's (NIST) computer security glossary[2] contains around ten thousand terms and definitions. From a corpus perspective, system logs, network flows, malware, software vulnerabilities, and threat models are examples of the specialized contents that distinguish cyber intelligent system development from other domains.

Recently large language models (LLMs) have transformed the landscape of natural language processing (NLP) and artificial intelligence (AI) by demonstrating unprecedented capabilities in a wide range of tasks (OpenAI 2023). LLMs have the potential to enable and assist cybersecurity work including cyber intelligence, incident analysis, vulnerability assessment, threat modeling, computer forensics, and control management. However, the application of LLMs in cybersecurity, a field characterized by domain-specific jargons, evolving threats, and complex operating environments, remains underexplored. To unlock the full potential of LLMs in cybersecurity, there is an urgent need for both a consistent and uniform approach assessing LLMs and a fine-tuning strategy catering to this specialized domain.

In this paper, we present CyberBench and CyberInstruct, two innovative tools designed to address the challenges of applying LLMs in the field of cybersecurity. CyberBench is a multi-task benchmark that assesses LLMs' performance for NLP jobs related to cybersecurity, offering valuable insights into their strengths and weaknesses. By systematically evaluating various mainstream LLMs, CyberBench contributes to identifying areas for improvement and fosters the development of more effective models for cybersecurity applications. CyberInstruct, on the other hand, is a family of fine-tuned generative LLMs based on state-of-art open LLMs, where a training corpus from CyberBench is leveraged for creating specialized models with enhanced capabilities in the cybersecurity domain. By employing instruction-tuning, CyberInstruct achieves comparable results with GPT-4 and outperforms open LLM baselines in various cybersecurity tasks, as shown in the Figure 1, demonstrating the effectiveness of our fine-tuning approach.

The main contributions of this paper are as follows:

- Development and presentation of CyberBench, a first-published multi-task benchmarking framework for assessing the performance of generative LLMs in cybersecurity-related tasks, with collection and preprocessing of diverse and representative datasets from various cybersecurity sources.
- Evaluation of leading LLMs using CyberBench with appropriate evaluation metrics, providing insights into their capabilities and limitations in the cybersecurity context

[*] These authors contributed equally.

[1] https://en.wikipedia.org/wiki/Category:Computer_security

[2] https://csrc.nist.gov/glossary

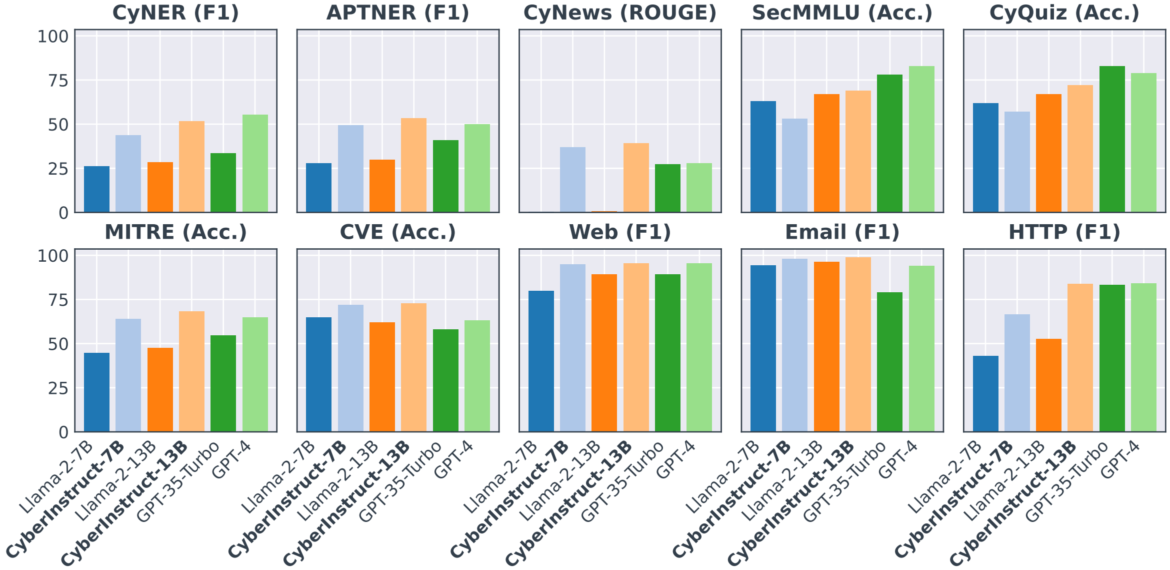

Figure 1: Evaluation results of baselines (Llama-2-7B, Llama-2-13B, GPT-35-Turbo, and GPT-4) and fine-tuned models (CyberInstruct-7B and CyberInstruct-13B) with 10 tasks from CyberBench, where the `all-mpnet-base-v2` embedding model is used for retrieving five-shot similar examples besides the summarization task with the zero-shot setting. CyNER and APTNER are named entity recognition (NER) tasks, CyNews is a summarization task, SecMMLU and CyQuiz are multiple-choice tasks, and the rest of datasets are text classification tasks. The F1, ROUGE, and accuracy (Acc.) scores are used as evaluation metrics for different tasks.

for named-entity recognition, summarization, multiple choice, and text classification tasks.

- Development of a specialized LLM fine-tuned for cybersecurity tasks, CyberInstruct, and demonstration of the effectiveness of instruction-tuning with parameter-efficient fine-tuning (PEFT), including quantization and low-rank adaptation, in improving LLMs' performance in the cybersecurity domain.

The paper is organized as follows: Section 2 reviews related work about benchmarks and language models. Section 3 presents the design and development of CyberBench. Section 4 introduces foundation model and fine-tuning methodology of CyberInstruct. Section 5 describes the experimental setup and results. Finally, Section 6 concludes the paper.

2 Related Work

The application of pretrained language models (PLMs) including large language models (LLMs) in specialized domains has been an active area of research in recent years. Many studies (Ling et al. 2023) have explored the effectiveness of these models in various fields. In this section, we briefly review related work in the context of benchmarking PLMs and domain-specific models in cybersecurity.

2.1 Language Model Benchmarks

Numerous benchmarks have been developed to assess the performance of general-purpose and domain-specific language models across a wide range of natural language processing (NLP) tasks. Some of the most notable benchmarks for general language models are GLUE (General Language Understanding Evaluation) (Wang et al. 2018), SuperGLUE (Wang et al. 2019), GLGE (General Language Generation Evaluation) (Liu et al. 2021), MMLU (Measuring Massive Multitask Language Understanding) (Hendrycks et al. 2020), and HELM (Holistic Evaluation of Language Models) (Liang et al. 2022). Also, in the privacy domain, there are PLUE (Privacy Policy Language Understanding Evaluation) (Chi et al. 2023) and PrivacyGLUE (Shankar et al. 2023). While these benchmarks have been instrumental in evaluating general-purpose and privacy-specific language models, they do not address the unique challenges and requirements of the cybersecurity domain. Scattered downstream NLP tasks have been used for evaluating language models in the cybersecurity domain, such as sentiment analysis and named-entity recognition (NER) for SecureBERT (Aghaei et al. 2022), NER and text classification for CyBERT (Ranade et al. 2021), clustering, word similarity, NER, text classification, and SuperGLUE for CySecBERT (Bayer et al. 2022), and multiple-choice questions in SecQA (Liu 2023). However, there is no systematical way to evaluate all cybersecurity language models, especially generative LLMs. Although there are some related works in progress such as Skyhawk Security[3] and Sophos AI[4], the lack of a

[3] https://skyhawk.security/new-horizons-in-cloud-security-part-1/, https://skyhawk.security/new-horizons-in-cloud-security-part2/

[4] https://news.sophos.com/en-us/2023/10/18/sophos-ai-team-to-present-at-camlis/

published benchmark tailored for language models in cybersecurity has left a gap in the evaluation and comparison of models designed for this domain.

CyberBench aims to fill this gap by providing a multi-task evaluation platform focused on cybersecurity. Instead of a comprehensive counterpart to HELM and other general benchmarks, CyberBench concentrates on NLP tasks related to this specific domain. It allows researchers and practitioners to assess the performance of language models on a range of cybersecurity tasks, facilitating the development and fine-tuning of models that can effectively address the challenges faced in this field. By creating a cybersecurity-specific benchmark, the research community can foster more targeted advancements in language understanding and generation for cybersecurity applications.

2.2 Domain-Specific Language Models

In the past few years, many language models have been developed or adapted for specific domains to address the unique challenges faced in those areas. In the context of cybersecurity, previous language models often focus on single or highly related tasks such as detecting anomalous logs (Guo, Yuan, and Wu 2021; Le and Zhang 2021; Ott et al. 2021; Liu and Buford 2023) and malicious code (Rahali and Akhloufi 2021, 2023), identifying vulnerabilities (Das et al. 2021), automating incident response (Shahid and Debar 2021; Ameri et al. 2021), and generating human-readable summaries of security events (Meng et al. 2023), while some previous works (Aghaei et al. 2022; Ranade et al. 2021; Bayer et al. 2022) pretrained or fine-tuned language models for border cybersecurity tasks. Some examples are SecureBERT (Aghaei et al. 2022), CyBERT (Ranade et al. 2021), CySecBERT (Bayer et al. 2022), MalBERT (Rahali and Akhloufi 2021), ThreatCrawl (Kuehn, Schmidt, and Reuter 2023), and V2W-BERT (Das et al. 2021).

Despite significant advances in cybersecurity NLP research using BERT-based models, there is still a need for a more versatile and user-guided solution. Those previous models are either proposed for one specific application or needed fine-tuning for a single task, which limit their usage for handling multiple cybersecurity tasks simultaneously and switching tasks based on user instructions. CyberInstruct addresses this need as a generative model designed for multi-task performance in the cybersecurity domain. Unlike previous work, CyberInstruct leverages instruction tuning, allowing it to follow user instructions and adapt to a wide range of cybersecurity tasks. This added versatility empowers users to obtain tailored results and insights, making CyberInstruct a valuable asset in addressing the growing challenges and complexities of the cybersecurity landscape.

In conclusion, the related work discussed in this section highlights the ongoing research efforts in the development and application of LLMs and benchmarks. CyberBench and CyberInstruct contribute to these research areas by offering tools and methodologies specifically designed for the cybersecurity domain, paving the way for future advancements and applications of LLMs in this critical field.

3 CyberBench

In this section, we propose a benchmark for evaluating large language models (LLMs) within the cybersecurity domain. We first discuss the main principles guiding the benchmark's design and then present tasks and datasets included. We also introduce instructions, prompts, and evaluation metrics.

3.1 Design Principles

CyberBench is a multi-task cybersecurity benchmark designed to evaluate the performance and effectiveness of large language models (LLMs) for cybersecurity-related tasks in English. In designing such benchmark for evaluating LLMs in the cybersecurity domain, several key principles are taken into consideration. Primarily, the benchmark should include diverse and representative tasks that can capture the complexity and nuances of the cybersecurity domain. This ensures that the evaluated models can handle various types of tasks relevant to the cybersecurity. Furthermore, the chosen tasks should ideally be based on widely accepted standards and datasets, allowing for reliable comparisons between different models and fostering research reproducibility.

With these principles in mind, ten datasets from four tasks are selected for the benchmark: named-entity recognition (NER), summarization (SUM), multiple choice (MC), and text classification (TC). Named-entity recognition is crucial in the cybersecurity domain as it enables the extraction of key information such as threat actors, vulnerabilities, and attack methods from unstructured text data. Summarization helps cybersecurity professionals in condensing lengthy cybersecurity reports or articles into concise and relevant information, allowing quicker and more efficient decision-making by security analysts. Multiple choice tasks test the model's ability to reason and draw logical conclusions from given information, a vital skill when dealing with complex cybersecurity incidents. Lastly, text classification is essential for organizing and categorizing the vast amounts of textual data generated in the cybersecurity domain, which in turn, facilitates efficient information retrieval and analysis. By incorporating these four kinds of tasks and corresponding datasets, CyberBench is created as a multi-task framework for evaluating the performance of LLMs in the context of cybersecurity.

3.2 Tasks and Datasets

A brief summary of CyberBench is presented in the Table 1. Examples from these datasets are shown in the Appendix A. The method for downloading the datasets is described in the Appendix B. These tasks and datasets are introduced in the following paragraphs.

Named-Entity Recognition The named-entity recognition (NER) is to recognize the entities from a list of given entity types. This task is crucial for cybersecurity, which allows the automatic identification and classification of cyber entities, such as exploits, security organization, vulnerability indicators, and so on from unstructured texts. NER helps professionals analyze textual data to detect potential threats and automate incident response, thereby strengthening cy-

Dataset	Data	Train	Val	Test	Input	Output	Metric
				Named-Entity Recognition (NER)			
CyNER	4,017	2,558	762	697	sentence	entities	micro F1
APTNER	9,971	6,923	1,669	1,379	sentence	entities	micro F1
				Summarization (SUM)			
CyNews	3,742	2,993	374	375	article	headline	ROUGE-1/2/L
				Multiple Choice (MC)			
SecMMLU	116	5	11	100	question and choices	answer	accuracy
CyQuiz	128	5	23	100	question and choices	answer	accuracy
				Text Classification (TC)			
MITRE	10,873	8,698	1,087	1,088	procedure description	technique ID and name	accuracy
CVE	14,652	11,721	1,465	1,466	CVE description	severity	accuracy
Web	11,429	9,143	1,143	1,143	URL	phishing or legitimate	binary F1
Email	13,281	10,624	1,328	1,329	email	phishing or safe	binary F1
HTTP	12,213	9,770	1,221	1,222	HTTP requests	anomalous or normal	binary F1

Table 1: Tasks and Datasets in the CyberBench with their whole data sizes, training (Train), validation (Val), and testing (Test) data sizes, input and output text types, and evaluation metrics.

ber defense capabilities. In CyberBench, we consider two datasets: CyNER and APTNER.

Cybersecurity NER (CyNER) (Alam et al. 2022) is the dataset of an open-source python library for NER in the cybersecurity domain. This dataset was manually extracted, cleaned, and annotated with 5 entities: malware, indicator, system, organization, and vulnerability . This dataset was already split into training, validation, and testing sets based on documents. During preprocessing, we split paragraphs into sentences and drop single-token sentences. We also fix tag issues and drop duplicates during data cleaning. The BIO-tags of the dataset have been transformed into a JSON format: ”entity type”: [”entity 1”, ”entity 2”, ...], which can be handled conveniently by generative LLMs.

Advanced Persistent Threat NER (APTNER) (Wang et al. 2022) provides a dataset for the NER in cyber threat intelligence research. This dataset contains 21 entities, from threat participants, security teams, to URLs, domains, and hash values, and was manual annotated by trained students and validated by professional people. The data was split as 7:1.5:1.5 by authors. During our preprocessing, we split paragraphs into sentences and drop single-token sentences as CyNER. We also fix tag issues with heuristic rules and drop duplicates. Similar to the CyNER, the output is formatted into a JSON text.

Summarization Text summarization (SUM) is the process of condensing large pieces of text into shorter, coherent summaries while preserving core information. In cybersecurity, this technique enables analysts to efficiently extract key points from extensive logs and reports, streamlining incident response and threat detection. For CyberBench, one public cybersecurity news dataset, CyNews, is selected.

Cybersecurity News Article Dataset (CyNews) (Ahmed et al. 2021) contains cybersecurity news from the Hacker News®. The original data classified the news articles into several types of cyber threats. In our benchmark, we use its news articles as the model input and headlines as the model output to transform it into a summarization task.

Multiple Choice Multiple-choice (MC) question answering involves selecting the correct answer from a list of options based on a given context or query. In cybersecurity, this task can evaluate the model knowledge about the domain, which is useful for answering basic questions during profession daily uses and employee training. For CyberBench, we decide two datasets: SecMMLU and CyQuiz.

MMLU Computer Security (SecMMLU) (Hendrycks et al. 2020) is a subset of MMLU (Measuring Massive Multitask Language Understanding) in the computer security domain. The original MMLU consists multiple-choice questions from various fields, and only the computer security domain is used in this research. This dataset does not have a training set for the computer security questions, besides a development set with 5 examples for the few-shot setting. This would require that the model should already learn the domain knowledge during the pre-training, rather than learn that knowledge from the MMLU few-shot examples or the training set.

Cybersecurity Skill Assessment[5] (CyQuiz) is the cybersecurity subset of the practice questions for professional skill assessments. These practice questions can be used for evaluating the candidates' abilities of general knowledge about the cybersecurity domain. In this benchmark, we only keep the multiple-choice questions with 4 choices and single correct answer. As the setting of MMLU, we keep 5 examples in the training set, 100 examples in the testing set, and put other examples in the validation set.

[5] https://github.com/Ebazhanov/linkedin-skill-assessments-quizzes

Text Classification Text classification (TC) is the task of assigning predefined categories or labels to a given text based on its content. In cybersecurity, this technique is widely used for many applications, such as phishing email and website detection, log anomaly detection, threat analysis, and malware classification, which can help analysts to efficiently prioritize potential threats. Although many public datasets available for this task in the cybersecurity domain, we select five representative datasets from five different cybersecurity areas: MITRE, CVE, Web, Email, and HTTP.

MITRE ATT&CK® Tagging[6] (MITRE) is a dataset collected from the MITRE ATT&CK® framework, which a knowledge base for cyber adversarial actions. The sentence-label pairs are extracted from procedure examples of each technique in this framework, where the input is a description of one procedure example, and the output is the technique ID and name. The procedure here is a specific implementation of a technique or sub-technique. During preprocessing, we drop the sentences with multiple labels and techniques with less than 10 examples.

CVE® and CWE™ Mapping Dataset[7] (CVE) is a dataset collected from National Vulnerability Archive (NVD) for publicly disclosed software vulnerabilities. In this research, the vulnerability description is used as the textual input and the severity level (critical, high, medium, and low) are used as labels. Due to the large amount of data, we only keep the CVEs from the year of 2021. We also drop duplicates and long inputs with more than 2,000 characters.

Webpage Phishing Detection[8] (Web) is a dataset with URLs and extracted features for building and benchmarking phishing detection systems. The features were extracted from URLs and contents of webpages. This dataset is balanced with half legitimate URLs and half phishing ones. In CyberBench, we use only URLs without extra features for the URL classification task.

Phishing Email Detection (Email) (Chakraborty 2023) provides a dataset for detecting phishing emails with the email texts. This dataset has 61% safe emails and 39% phishing emails. During preprocessing, we drop long emails with more than 2,000 characters.

HTTP Dataset CSIC 2010 (HTTP) (Giménez, Villegas, and Marañón 2010) is dataset containing web requests automatically generated and targeted to an e-commerce web application. This dataset was developed for evaluating web attack protection systems. Each HTTP request can have multiple fields, including the method, user agent, cache control, etc. The anomalous class includes three types of malicious requests: static attack, dynamic attack, and unintentional illegal request. Due to the large amount of data, we use only 20% of original test sets from both normal and anomalous classes for a balanced dataset.

3.3 Instructions and Prompts

To evaluate large language models (LLMs) using CyberBench, it is essential to provide clear instructions for each task. Instead of one human-crafted instruction for one dataset, 10 distinct instructions are generated with GPT-4 for each dataset and randomly assigned to examples. This design can increase the robustness of benchmark and avoid introducing too much dependence on single instruction. For named-entity recognition (NER) tasks, the instructions encompass entity types, entity definitions, and the desired output format. In the case of text classification tasks, the instructions specify the input text type and output categories. The Appendix A showcases examples of these instructions. During the experiments, the instruction, input, and output from each example are formatted into a single prompt as the Alpaca (Taori et al. 2023) instruction tuning dataset.

To enhance model performance and encourage the generation of outputs in the specified format, we employ a few-shot prompt approach with Retrieval-Augmented Generation (RAG) (Lewis et al. 2020), where examples are selected from the training set by using an example selector (Chase 2022) based on similarity search. Inside the selector, examples are embedded into vectors using an embedding model, and the similarity scores between the given test example and the training examples are calculated. The examples with the highest similarity scores are selected by utilizing the nearest neighbor search in a vector store (Johnson, Douze, and Jégou 2019). The five-shot setting is applied across all tasks, except for summarization, where only the zero-shot setting is applied due to the size limit of context window. The few-shot prompt template is shown in the Appendix C.

3.4 Evaluation Metrics

Evaluating the performance of large language models (LLMs) on the selected datasets necessitates the use of appropriate evaluation metrics. We apply suitable evaluation metrics for different task in the CyberBench and introduce them as follows.

For the named-entity recognition (NER) task, entities can be generated for all entity types. Therefore, micro-averaged F1 scores are employed by counting all entity types. In the case of the summarization task, the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin 2004) scores, including ROUGE-1, ROUGE-2, and ROUGE-L, are utilized, where ROUGE-1 counts the overlap of unigrams between predictions and references, ROUGE-2 counts the bigrams, and ROUGE-L is computed based on the longest common substring (LCS). For multiple choice tasks, the accuracy serves as the metric. And for the text classification tasks, accuracy is applied to multi-class tasks (more than two classes), whereas the F1 score for the positive class (e.g., phishing or anomalous) is used for binary classification tasks. To compute the average score for the ten datasets, a simple averaging method is employed. However, for summarization tasks, the average of ROUGE-1, ROUGE-2, and ROUGE-L scores is calculated first before deriving the overall average. This approach ensures a comprehensive evaluation of LLM performance across a diverse range of tasks and datasets.

[6] https://attack.mitre.org/

[7] https://www.kaggle.com/datasets/krooz0/cve-and-cwe-mapping-dataset

[8] https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset

Using CyberBench, we can evaluate the performance of several leading LLMs on the defined cybersecurity tasks. The results of these evaluations will be presented in the experiment section. Before discussing experiment results, we will introduce a family of fine-tuned LLMs for cybersecurity and its methods in the next section.

4 CyberInstruct

CyberInstruct is a family of fine-tuned generative large language models (LLMs) designed to address the unique challenges and requirements inherent to cybersecurity domain. In this section, we introduce the foundation model, instruction tuning, and parameter-efficient fine-tuning (PEFT).

4.1 Foundation Model

Foundation models, also referred to as base models, represent a class of large machine learning models, characterized by training on extensive datasets at scale and possessing the remarkable ability to adapt to a broad spectrum of downstream tasks. In the field of natural language processing (NLP), LLMs can be served as foundation models for applications in specific domains. Among numerous open LLMs presented in the past few years, Llama-2 is a recently introduced collection of pretrained and fine-tuned large language models, specifically designed to cater to a variety of use cases, with a particular focus on dialogue applications. Given the impressive performance and adaptability of Llama-2, it serves as an excellent foundation model for fine-tuning in the context of cybersecurity. By building upon Llama-2's optimized training and robust performance, we can create a family of fine-tuned models, CyberInstruct, which can effectively tackle the complex challenges of the cybersecurity domain using NLP techniques.

4.2 Instruction Tuning

The fine-tuning process (Chung et al. 2022; Longpre et al. 2023) of CyberInstruct involves leveraging the training datasets from CyberBench to create a specialized model with enhanced capabilities in the cybersecurity domain. As described in the benchmark section, the CyberBench dataset encompasses a diverse range of cybersecurity data sources. To outperform baseline LLMs, CyberInstruct employs a technique known as instruction-tuning. This approach involves incorporating explicit instructions into the model's input during the fine-tuning process, which guides the model to generate more accurate and relevant outputs for cybersecurity tasks. This is particularly important for domain-specific tasks, where the general knowledge of LLMs may not be sufficient to achieve high performance. A key difference of instruction tuning used in this research from the traditional fine-tuning for pretrained language models such as BERT is that we only fine-tune a single model for all tasks in the CyberBench. There, CyberInstruct can handle multitasks at the same time without the need for additional fine-tuning or separate models for each task.

In preparation for the evaluation tasks, we format all examples into instructions, inputs, and outputs, and combine them into texts by using the Alpaca (Taori et al. 2023) prompt template. However, we only use the zero-shot prompt template here without few-shot examples. For each input paired with the instruction, the output is also included in the text to teach the model how to handle the input properly.

4.3 Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) (Liu et al. 2022) is a pivotal approach for tailoring large language models (LLMs) to specialized tasks or domains with the goal of minimizing the number of trainable parameters. This approach strikes a balance between model performance and resource consumption. In the development of CyberInstruct, we employ a method called QLoRA, Quantized Low-Rank Adaptation (Dettmers et al. 2023), where gradients backpropagate through frozen 4-bit quantized pre-trained layers with trainable low rank adapters (Hu et al. 2021) applied on top of selected attention modules. By employing QLoRA for fine-tuning the CyberInstruct, we aim to optimize the utilization of the extensive knowledge encapsulated within the pre-trained LLM, while tailoring it to the specific needs of the cybersecurity domain. This method enables us to achieve superior performance on the benchmark tasks while maintaining efficient resource utilization. Consequently CyberInstruct emerges as a potential tool for tackling a broad spectrum of cybersecurity challenges.

5 Experiments

In this section, we introduce baselines and experiment setup and then delineate the experiment results obtained from evaluating large language models (LLMs) and CyberInstruct using our benchmark suite CyberBench.

5.1 Baselines and Experiment Setup

In this study, we select a collection of pretrained language models (PLMs) to evaluate their performance against our cybersecurity benchmark, CyberBench. These baselines are drawn from two prominent groups of models in the field of natural language processing (NLP) and cybersecurity. The first group consists of BERT-based models that have been specifically pretrained for the cybersecurity domain, while the second group encompasses various large language models (LLMs) that showcase a range of architectures and capabilities.

The group of BERT-based models (Devlin et al. 2018) pretrained for cybersecurity includes SecBERT[9], SecRoBERTa[10], SecureBERT (Aghaei et al. 2022), and CySecBERT (Bayer et al. 2022). These models are derivatives of the original BERT and RoBERTa architectures and have been pretrained or fine-tuned on cybersecurity-related corpora to enhance their performances in domain-specific tasks. The second set of baselines used in this study includes generative LLMs: Falcon (Penedo et al. 2023; Almazrouei et al. 2023), Vicuna (Zheng et al. 2023), Mistral (Jiang et al. 2023), Zephyr (Tunstall et al. 2023), Llama-2 (Touvron et al.

[9] https://huggingface.co/jackaduma/SecBERT

[10] https://huggingface.co/jackaduma/SecRoBERTa

2023), GPT-3.5 (Ouyang et al. 2022), and GPT-4 (OpenAI 2023). These models represent various advancements in the field of NLP and demonstrate the potential of LLMs in tackling intricate language understanding and generation tasks. In this work, we select open LLM variants with 7 billion and 13 billion parameters.

To evaluate the performance of LLMs using CyberBench and validate the effectiveness of CyberInstruct, we conducted a series of experiments. During experiments with generative LLMs, the temperature is set to 0. To reduce the text generation time and cost, we truncate the model response at the end of the first output line, which is suitable for all tasks in the CyberBench by design. Model responses are then checked for matching the correct outputs, and evaluation metrics are subsequently calculated. The BERT-based models are fined-tuned for each task. The foundation models are also fine-tuned with the CyberBench corpus for producing CyberInstruct. More details about evaluation and fine-tuning setup can be found in the Appendix D.

5.2 Experiment Results

In this part, we present evaluation results for both baselines and fine-tuned models on CyberBench. More discussions about the effects of embedding model, few-shot examples, and quantization for model performances can be found in the Appendix E.

Results of Baselines The experiment results for baselines are show in the Table 2. Here we separate the BERT-based models and the generative LLMs given the distinct operational characteristics of each. BERT-based models require finetuning for each task, whereas generative LLMs can perform multiple tasks without the need for fine-tuning. The `all-mpnet-base-v2` embedding model (Reimers and Gurevych 2019; Song et al. 2020) is used for retrieving few-shot examples for LLMs. In terms of overall performance, GPT-4 outperforms all other LLMs over this benchmark, while GPT-3.5 follows closely, with Mistral-7B and Zephyr-7B trailing behind.

For NER tasks, the BERT-based models outperform most generative LLMs, while the GPT-4 can achieve comparable performances with SecBERT and SecRoBERTa. For the summarization task, GPT-4 and Vicuna-7B achieve the highest scores but are followed by GPT-3.5 and Vicuna-13B approximately. For multiple-choice tasks, GPT-4 tops the SecMMLU dataset, while GPT-3.5 excels in the CyQuiz dataset. For the five text classification tasks, the BERT-based models achieve better performances than generative LLMs. Among the generative models, GPT-4 obtains the highest scores for MITRE, Web, and HTTP datasets. Llama-2-7B and Mistral-7B perform well in the CVE task, and Llama-2-13B and Mistral-7B are the best models for the Email dataset.

These evaluation results underscore the robust performances of GPT-4 in cybersecurity tasks. However, due to the proprietary nature of OpenAI models, it is expensive to call OpenAI APIs and impose many limitations about using sensitive or confidential data, which are common scenarios in the cybersecurity domain. These constraints provide spaces for fine-tuning open LLMs in the cybersecurity domain.

Results of Fine-Tuned Models To demonstrate the effectiveness of our fine-tuning strategy, we evaluate the performance of CyberInstruct-7B and CyberInstruct-13B, both fine-tuned with QLoRA, on the cybersecurity tasks defined in the CyberBench. We compare these results with the foundation models and the previously best performing baselines, GPT-3.5 and GPT-4. The evaluation results are presented in the Table 2 and Figure 1.

Two training datasets are used during the fine-tuning Llama-2. The first is the training set of CyberBench, and the second is a subset of training data from MMLU (Hendrycks et al. 2020). Given the scarcity of training data for multiple-choice questions in the cybersecurity domain, we leverage science questions of elementary and middle difficulties from MMLU to enhance CyberInstruct performance in the multiple-choice tasks, SecMMLU and CyQuiz.

The results show that CyberInstruct-13B, fine-tuned Llama-2-13B with both CyberBench training set and the MMLU science questions, achieves the best overall evaluation result. The fine-tuned Llama-2-13B with only the CyberBench training set demonstrates the comparable performance to GPT-4. Those fine-tuned models show their strengths in text classification and summarization tasks within the cybersecurity domain. However, even with training data of multiple-choice questions from science domain, CyberInstruct-13B is still inferior to GPT models. In the NER tasks, fine-tuned Llama-2-13B models have comparable performance with GPT-4.

In summary, CyberInstruct presents a promising approach for adapting LLMs to the cybersecurity domain. By integrating instruction-tuning with parameter-efficient fine-tuning, it creates a specialized model with superior performance in cybersecurity tasks. The development of CyberInstruct paves the way for future research and the application of LLMs in the cybersecurity field, enabling researchers and practitioners to better tackle the complex challenges of this critical domain.

6 Conclusion

In this research, we present CyberBench and CyberInstruct, two innovative tools designed to tackle the challenges of applying large language models (LLMs) in the cybersecurity domain. CyberBench, a multi-task benchmark, systematically evaluates the performance of LLMs across cybersecurity-related tasks, including named entity recognition, summarization, multiple choice, and text classification, providing valuable insights into their strengths and weaknesses in this specialized domain. The development of CyberBench contributes to the identification of areas for improvement and fosters the advancement of more effective models for cybersecurity applications.

CyberInstruct, a family of fine-tuned generative LLMs, leverages the training sets from CyberBench to create a specialized model with enhanced capabilities in the cybersecurity domain. By employing instruction-tuning with parameter-efficient fine-tuning (PEFT), CyberInstruct achieves superior and comparable performance to baseline

Model	Average	CyNER	APTNER	CyNews	SecMMLU	CyQuiz	MITRE	CVE	Web	Email	HTTP
-	-	F1	F1	R-1/2/L	Acc.	Acc.	Acc.	Acc.	F1	F1	F1
BERT-based models											
SecBERT	-	49.8	53.2	-	-	-	80.2	74.6	94.9	98.0	87.6
SecRoBERTa	-	50.5	51.7	-	-	-	81.9	72.4	94.4	97.3	90.0
SecureBERT	-	72.5	61.1	-	-	-	84.6	75.7	96.4	98.5	92.0
CySecBERT	-	69.4	57.1	-	-	-	85.0	76.7	96.0	99.2	92.5
Generative LLMs – 7B											
Falcon-7B	39.4	24.1	17.7	1.0/0.8/1.0	27.0	27.0	34.9	54.6	68.9	93.3	45.2
Falcon-7B-Instruct	37.5	20.4	19.1	7.2/2.7/6.0	25.0	21.0	30.4	52.9	59.5	93.5	48.3
Vicuna-7B-v1.5	53.0	25.8	27.5	36.1/15.9/31.2	64.0	66.0	43.5	60.0	75.3	86.4	53.7
Mistral-7B-v0.1	58.1	36.7	33.0	3.4/1.7/3.0	76.0	77.0	50.2	64.6	91.9	96.4	52.6
Mistral-7B -Instruct-v0.1	55.0	32.3	26.2	28.7/11.8/24.5	72.0	69.0	47.3	58.7	87.2	88.9	47.2
Zephyr-7B-beta	57.7	30.0	30.5	32.0/12.8/27.4	74.0	75.0	43.5	61.9	85.2	86.7	66.2
Llama-2-7B	50.6	26.3	28.0	0.3/0.3/0.3	63.0	62.0	44.6	64.7	79.9	94.2	42.8
Llama-2-7B-Chat	44.6	22.7	25.4	25.2/9.6/21.6	60.0	56.0	41.6	52.5	48.4	79.4	41.0
Generative LLMs – 13B											
Vicuna-13B-v1.5	57.3	26.2	28.1	35.6/15.6/30.9	66.0	74.0	47.3	62.3	82.6	86.5	72.3
Llama-2-13B	54.1	28.6	29.9	0.6/0.5/0.6	67.0	67.0	47.5	62.1	89.3	96.4	52.5
Llama-2-13B-Chat	45.0	27.5	28.2	3.5/1.3/2.9	64.0	65.0	42.7	42.0	58.8	70.3	48.5
GPTs											
GPT-35-Turbo	62.6	33.4	40.9	35.5/15.4/30.3	78.0	**83.0**	54.5	58.0	89.2	78.9	83.1
GPT-4	69.6	**55.4**	50.0	35.9/15.5/31.2	**83.0**	81.0	64.9	63.0	95.4	93.9	**84.1**
Fine-tuned models with CyberBench training data											
Llama-2-7B	60.2	39.1	46.1	44.6/25.5/41.1	44.0	45.0	62.6	72.3	94.4	97.7	64.1
Llama-2-13B	69.9	53.3	53.3	**47.4**/27.2/43.2	71.0	62.0	**70.9**	**72.8**	**96.2**	98.5	82.1
Fine-tuned models with CyberBench training data and MMLU science questions (CyberInstruct)											
Llama-2-7B	63.5	43.9	49.5	44.0/25.2/40.5	53.0	57.0	63.8	72.0	94.8	97.9	66.6
Llama-2-13B	**70.4**	51.7	**53.4**	47.3/**27.5**/**43.3**	69.0	72.0	68.3	72.6	95.3	**98.6**	83.7

Table 2: Evaluation results for baselines and fine-tuned models on CyberBench, where the `all-mpnet-base-v2` embedding model is used for retrieving five-shot similar examples for LLMs besides the summarization task. The baselines are split into three groups based on the model architecture and number of parameters. Two kinds of training data are used, one from the CyberBench training set, and another from the MMLU training data of elementary and middle science questions. In evaluation metrics, "R" is the ROUGE score, and "Acc." is the accuracy.

LLMs across various cybersecurity tasks. This demonstrates the effectiveness of our fine-tuning approach and highlights the importance of domain-specific knowledge in achieving high performance in specialized tasks.

The development of CyberBench and CyberInstruct paves the way for future research and the application of LLMs in the cybersecurity field, enabling researchers and practitioners to better address the complex challenges of this critical domain. While CyberInstruct demonstrates promising results, it is essential to acknowledge its limitations and identify potential future research directions, such as improving model interpretability, ensuring data diversity and representation, addressing ethical considerations, and enhancing model robustness and generalizability.

In conclusion, the development of CyberBench and CyberInstruct marks critical progress towards the effective application of LLMs in the cybersecurity domain. By addressing the unique challenges and requirements of this field, these tools contribute to the ongoing research in adapting LLMs for specialized domains and pave the way for future advancements in the field of cybersecurity.

Acknowledgments

We would like to extend our gratitude to Andrew R. Plummer for his valuable suggestions and insights on benchmarking large language models in the cybersecurity domain. His contributions significantly enhanced the quality of our paper.

References

Aghaei, E.; Niu, X.; Shadid, W.; and Al-Shaer, E. 2022. SecureBERT: A Domain-Specific Language Model for Cybersecurity. In *International Conference on Security and Privacy in Communication Systems*, 39–56. Springer.

Ahmed, F.; Anwar, T.; Tanvir, S.; Saha, R.; Shoumo, S.; Hossain, S.; and Rasel, A. 2021. Cybersecurity News Article Dataset; Mendeley Data, V1.

Alam, M. T.; Bhusal, D.; Park, Y.; and Rastogi, N. 2022. CyNER: A Python Library for Cybersecurity Named Entity Recognition. *arXiv preprint arXiv:2204.05754*.

Almazrouei, E.; Alobeidli, H.; Alshamsi, A.; Cappelli, A.; Cojocaru, R.; Debbah, M.; Goffinet, E.; Heslow, D.; Launay, J.; Malartic, Q.; Noune, B.; Pannier, B.; and Penedo, G. 2023. Falcon-40B: An Open Large Language Model with State-of-the-art Performance.

Ameri, K.; Hempel, M.; Sharif, H.; Lopez Jr, J.; and Perumalla, K. 2021. CyBERT: Cybersecurity Claim Classification by Fine-Tuning the BERT Language Model. *Journal of Cybersecurity and Privacy*, 1(4): 615–637.

Bayer, M.; Kuehn, P.; Shanehsaz, R.; and Reuter, C. 2022. CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain. *arXiv preprint arXiv:2212.02974*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language Models are Few-Shot Learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chakraborty, S. 2023. Phishing Email Detection.

Chase, H. 2022. LangChain. https://github.com/langchain-ai/langchain.

Chi, J.; Ahmad, W. U.; Tian, Y.; and Chang, K.-W. 2023. PLUE: Language Understanding Evaluation Benchmark for Privacy Policies in English. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 352–365. Toronto, Canada: Association for Computational Linguistics.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2022. Scaling Instruction-Finetuned Language Models. *arXiv preprint arXiv:2210.11416*.

Das, S. S.; Serra, E.; Halappanavar, M.; Pothen, A.; and Al-Shaer, E. 2021. V2W-BERT: A Framework for Effective Hierarchical Multiclass Classification of Software Vulnerabilities. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, 1–12. IEEE.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Giménez, C. T.; Villegas, A. P.; and Marañón, G. Á. 2010. HTTP Data Set CSIC 2010. *Information Security Institute of CSIC (Spanish Research National Council)*, 64.

Guo, H.; Yuan, S.; and Wu, X. 2021. LogBERT: Log Anomaly Detection via BERT. In *2021 international joint conference on neural networks (IJCNN)*, 1–8. IEEE.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Johnson, J.; Douze, M.; and Jégou, H. 2019. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3): 535–547.

Kuehn, P.; Schmidt, M.; and Reuter, C. 2023. ThreatCrawl: A BERT-based Focused Crawler for the Cybersecurity Domain. *arXiv preprint arXiv:2304.11960*.

Le, V.-H.; and Zhang, H. 2021. Log-based Anomaly Detection Without Log Parsing. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 492–504. IEEE.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474.

Liang, P.; Bommasani, R.; Lee, T.; Tsipras, D.; Soylu, D.; Yasunaga, M.; Zhang, Y.; Narayanan, D.; Wu, Y.; Kumar, A.; et al. 2022. Holistic Evaluation of Language Models. *arXiv preprint arXiv:2211.09110*.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Ling, C.; Zhao, X.; Lu, J.; Deng, C.; Zheng, C.; Wang, J.; Chowdhury, T.; Li, Y.; Cui, H.; Zhang, X.; Zhao, T.; Panalkar, A.; Cheng, W.; Wang, H.; Liu, Y.; Chen, Z.; Chen, H.; White, C.; Gu, Q.; Pei, J.; and Zhao, L. 2023. Domain Specialization as the Key to Make Large Language Models Disruptive: A Comprehensive Survey. *arXiv preprint arXiv:2305.18703*.

Liu, D.; Yan, Y.; Gong, Y.; Qi, W.; Zhang, H.; Jiao, J.; Chen, W.; Fu, J.; Shou, L.; Gong, M.; et al. 2021. GLGE: A New General Language Generation Evaluation Benchmark. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 408–420.

Liu, H.; Tam, D.; Muqeeth, M.; Mohta, J.; Huang, T.; Bansal, M.; and Raffel, C. A. 2022. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning. *Advances in Neural Information Processing Systems*, 35: 1950–1965.

Liu, Z. 2023. SecQA: A Concise Question-Answering Dataset for Evaluating Large Language Models in Computer Security. *arXiv preprint arXiv:2312.15838*.

Liu, Z.; and Buford, J. 2023. Anomaly Detection of Command Shell Sessions based on DistilBERT: Unsupervised and Supervised Approaches. *arXiv preprint arXiv:2310.13247*.

Longpre, S.; Hou, L.; Vu, T.; Webson, A.; Chung, H. W.; Tay, Y.; Zhou, D.; Le, Q. V.; Zoph, B.; Wei, J.; et al. 2023. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. *arXiv preprint arXiv:2301.13688*.

Loshchilov, I.; and Hutter, F. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Meng, W.; Zaiter, F.; Zhang, Y.; Liu, Y.; Zhang, S.; Tao, S.; Zhu, Y.; Han, T.; Zhao, Y.; Wang, E.; et al. 2023. LogSummary: Unstructured Log Summarization for Software Systems. *IEEE Transactions on Network and Service Management*.

OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Ott, H.; Bogatinovski, J.; Acker, A.; Nedelkoski, S.; and Kao, O. 2021. Robust and Transferable Anomaly Detection in Log Data using Pre-Trained Language Models. In *2021 IEEE/ACM international workshop on cloud intelligence (CloudIntelligence)*, 19–24. IEEE.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training Language Models to Follow Instructions with h=Human Feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Penedo, G.; Malartic, Q.; Hesslow, D.; Cojocaru, R.; Cappelli, A.; Alobeidli, H.; Pannier, B.; Almazrouei, E.; and Launay, J. 2023. The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. *arXiv preprint arXiv:2306.01116*.

Rahali, A.; and Akhloufi, M. A. 2021. MalBERT: Malware Detection using Bidirectional Encoder Representations from Transformers. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 3226–3231. IEEE.

Rahali, A.; and Akhloufi, M. A. 2023. MalBERTv2: Code Aware BERT-Based Model for Malware Identification. *Big Data and Cognitive Computing*, 7(2): 60.

Ranade, P.; Piplai, A.; Joshi, A.; and Finin, T. 2021. CyBERT: Contextualized Embeddings for the Cybersecurity Domain. In *2021 IEEE International Conference on Big Data (Big Data)*, 3334–3342. IEEE.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*.

Shahid, M. R.; and Debar, H. 2021. CVSS-BERT: Explainable Natural Language Processing to Determine the Severity of a Computer Security Vulnerability from its Description. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1600–1607. IEEE.

Shankar, A.; Waldis, A.; Bless, C.; Andueza Rodriguez, M.; and Mazzola, L. 2023. PrivacyGLUE: A Benchmark Dataset for General Language Understanding in Privacy Policies. *Applied Sciences*, 13(6): 3701.

Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T.-Y. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. *Advances in Neural Information Processing Systems*, 33: 16857–16867.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Tunstall, L.; Beeching, E.; Lambert, N.; Rajani, N.; Rasul, K.; Belkada, Y.; Huang, S.; von Werra, L.; Fourrier, C.; Habib, N.; et al. 2023. Zephyr: Direct Distillation of LM Alignment. *arXiv preprint arXiv:2310.16944*.

Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 3266–3280.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 353–355.

Wang, X.; He, S.; Xiong, Z.; Wei, X.; Jiang, Z.; Chen, S.; and Jiang, J. 2022. APTNER: A Specific Dataset for NER Missions in Cyber Threat Intelligence Field. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 1233–1238. IEEE.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*.

A CyberBench Examples

Examples of instructions, inputs, and outputs from the datasets in CyberBench are shown in the Table 3.

B CyberBench Datasets

The datasets in the CyberBench can be downloaded into a `data` folder with the following subfolders and files:

- `ner` (named-entity recognition)
  - `cyner`: `train.txt`, `valid.txt`, and `test.txt` from CyNER (https://github.com/aiforsec/CyNER) `dataset/mitre`
  - `aptner`: `APTNERtrain.txt`, `APTNERdev.txt`, and `APTNERtest.txt` from APTNER (https://github.com/wangxuren/APTNER)
- `sum` (summarization):
  - `cynews`: `TheHackerNews_Dataset.csv` from Cybersecurity-News-Article-Dataset (https://github.com/cypher-07/Cybersecurity-News-Article-Dataset) (saved to CSV)
- `mc` (multiple-choice):
  - `secmmlu`: `computer_security_dev.csv`, `computer_security_val.csv`, `computer_security_test.csv`, `science_elementary.csv`, and `science_middle.csv` from MMLU (https://huggingface.co/datasets/cais/mmlu) `data.tar`
  - `cyquiz`: `cybersecurity-quiz.md` from Skill-Assessments (https://github.com/Ebazhanov/linkedin-skill-assessments-quizzes) `cybersecurity`
- `tc` (text classification):
  - `mitre`: `enterprise-attack.json` from MITRE-CTI (https://github.com/mitre/cti) `enterprise-attack` (v13.1)
  - `cve`: `Global_Dataset.csv` from CVE-and-CWE-Mapping-Dataset (https://www.kaggle.com/datasets/krooz0/cve-and-cwe-mapping-dataset) (saved to CSV)
  - `web`: `dataset_phishing.csv` from Webpage-Phishing-Detection-Dataset (https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset)
  - `email`: `Phishing_Email.csv` from Phishing-Email-Detection (https://www.kaggle.com/datasets/subhajournal/phishingemails)
  - `http`: `normalTrafficTraining.txt`, `normalTrafficTest.txt`, and `anomalousTrafficTest.txt` from HTTP-Dataset-CSIC-2010 (https://www.tic.itefi.csic.es/dataset/)

C Few-Shot Prompt Template

The few-shot prompt template based on Alpaca (Taori et al. 2023) is shown below, where inputs and outputs are from few-shot examples, and [...] represents the omitted text:

```
Below is an instruction that describes a
task, paired with an input that provides
further context. Write a response that
appropriately completes the request.

### Instruction:
{instruction}

### Input:
{example_input_1}

### Response:
{example_output_1}

[...]

### Input:
{example_input_n}

### Response:
{example_output_n}

### Input:
{input}

### Response:
```

D Experiment Setup Details

In addition to the general setting for generative LLMs, specific settings are applied for each task. For NER, we eliminate duplicated entities within each entity type, and check for matched entities in that type. The matched entities from all entity types are accumulated for computing the micro F1 score. For multiple-choice and text classification tasks, we check if the generated output can exactly match the true output, although the fuzzy matching can be implemented in future research.

The BERT-based models cannot directly generate responses for prompts due to the transformer encoder structure. Therefore, a task-specific header (neural network) is required for each task, and the whole model with additional layers needs fine-tuning. The BERT-based models are fined-tuned for each task with a batch size of 16, a learning rate of 5e-5, a linear learning rate scheduler, a warmup ratio of 0.02, and a weight decay of 0.01 with the AdamW (Loshchilov and Hutter 2018) optimizer. The NER task is treated as token classification with fine-tuning over 5 epochs, and the text classification task is treated as sequence classification with fine-tuning over 10 epochs. Due to the limitations of model structure as a transformer encoder and lack of training data of cyber multiple-choice questions, we do not evaluate the BERT-based models for the summarization and multiple-choice tasks.

During fine-tuning of CyberInstruct, the foundation model is transformed into 4-bit quantization, and the LoRA layers are added to four matrices (query, key, value, and output) in each attention module. A batch size of 16, a learning rate of 5e-5, a linear learning rate scheduler, a warmup ratio of 0.02, and a weight decay of 0.00 are used with the paged AdamW 8-bit optimizer.

E More Experiments and Analyses

E.1 Analysis of Embedding Models

To understand the impact of embedding models, we compare the GPT-3.5 and GPT-4 performances with two embedding models, `text-embedding-ada-002-2` (Brown et al. 2020) from OpenAI and `all-mpnet-base-v2` (Reimers and Gurevych 2019; Song et al. 2020) from Hugging Face. The results presented in the Table 4 indicate that the overall performance of each model is relatively consistent across the two embedding models besides few variances probably due to OpenAI API calls. For the convenience of experimenting, we only evaluate the remaining LLMs with the `all-mpnet-base-v2` embedding model in this research.

E.2 Analysis of Few-Shot Examples

To analyze the impact of few-shot examples on CyberBench, we compare the performances of Llama-2-7B and Llama-2-7B-Chat with one and five shots for retrieving similar or random examples. These results are presented in the Table 5. It is important to note that we only employed zero-shot setting for the summarization task. We refrained from using zero-shot setting for other tasks due to the free-format nature of the output without illustrative examples for the LLMs, which makes output parsing and evaluating challenging. The LLMs that utilized five examples from similarity search achieve the highest average scores, underscoring the effectiveness of Retrieval-Augmented Generation (RAG) (Lewis et al. 2020) in the cybersecurity domain. This is particularly evident in NER and text classification tasks, where similar examples can provide valuable contexts beyond the given instructions. For the remainder of our experiments, we only use the five-shot prompt with examples from similarity search.

E.3 Analysis of Quantization Precisions

To analyze the impact of quantization, we select two open-source LLMs, Llama-2-7B and Llama-2-7B-Chat, and conduct CyberBench evaluation tasks under 4-bit and 8-bit quantization settings. The evaluation results, presented in Table 6 do not show significant differences overall between these quantization precisions, although some variances are observed in specific tasks. These results confirm the benefits of employing quantization to limit computational costs while keeping the result accuracy.

Dataset	**Instruction**	**Input**	**Output**
Named-Entity Recognition (NER)			
CyNER	Within the provided sentence, find entities that correspond to these cybersecurity domain entity types: Malware, System, Organization, Indicator, Vulnerability. To assist you, here are the definitions of the entities: [. . .] Extract and arrange the entities in a JSON object according to this format: ”entity type”: [”entity 1”, ”entity 2”, ...]. Do not include entities that are not part of the sentence.	Super Mario Run Malware #2 – DroidJack RAT Gamers love Mario and Pokemon, but so do malware authors.	”Malware”: [”Super Mario Run Malware”, ”DroidJack RAT”], ”System”: [”Mario”, ”Pokemon”]
APTNER	Within the provided sentence, find entities that correspond to these cybersecurity domain entity types: APT, SECTEAM, IDTY, OS, EMAIL, [. . .]	From April 19-24, 2017, a politically-motivated, targeted campaign was carried out against numerous Israeli organizations.	”TIME”: [”April 19-24, 2017”], ”LOC”: [”Israeli”]
Summarization (SUM)			
CyNews	What would be a fitting headline for this text discussing recent advancements or incidents in cybersecurity?	Cloud infrastructure security company Wiz on Thursday revealed details of a now-fixed Azure Cosmos database vulnerability that could have been potentially exploited to grant any Azure user full admin access to other customers’ database instances without any authorization. [. . .]	Critical Cosmos Database Flaw Affected Thousands of Microsoft Azure Customers
Multiple Choice (MC)			
SecMMLU	Please assess the cybersecurity question and indicate the most suitable answer among the given choices.	Question: SHA-1 has a message digest of A. 160 bits B. 512 bits C. 628 bits D. 820 bits	A
CyQuiz	Considering the cybersecurity subject matter, pick the most accurate solution for the presented question.	Question: You are at a coffee shop and connect to a public wireless access point (WAP). What a type of cybersecurity attack are you most likely to experience? A. man-in-the-middle attack B. back door C. logic bomb D. virus	A
Text Classification (TC)			
MITRE	Examine the procedure example and ascertain the appropriate MITRE ATT&CK technique ID and name.	APT41 used a compromised account to create a scheduled task on a system.	T1053.005 Scheduled Task/Job: Scheduled Task
CVE	Based on the CVE description provided, determine the appropriate severity level: critical, high, medium, or low.	Improper conditions check in some Intel(R) Ethernet Controllers 800 series Linux drivers before version 1.4.11 may allow an authenticated user to potentially enable information disclosure or denial of service via local access.	high
Web	Examine the URL and categorize it as phishing or legitimate.	http://rgipt.ac.in	legitimate
Email	Identify if the given email is phishing or safe.	the other side of * galicismos * * galicismo * is a spanish term which names the improper introduction of french words which are spanish sounding and thus very deceptive to the ear . * galicismo * is often considered to be a * barbarismo * . [. . .]	safe
HTTP	Evaluate the HTTP request below and classify it as either normal or anomalous.	GET [. . .] HTTP/1.1 User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko) Pragma: no-cache Cache-control: no-cache [. . .]	anomalous

Table 3: Examples of instructions, inputs, and outputs from the datasets in CyberBench, where [...] represents the omitted text.

Model	**Average**	**CyNER**	**APTNER**	**CyNews**	**SecMMLU**	**CyQuiz**	**MITRE**	**CVE**	**Web**	**Email**	**HTTP**
-	-	F1	F1	R-1/2/L	Acc.	Acc.	Acc.	Acc.	F1	F1	F1
Embedding model: `text-embedding-ada-002-2`											
GPT-35-Turbo	61.9	28.2	40.0	35.5/15.4/30.3	80.0	82.0	55.5	61.1	87.0	77.6	80.3
GPT-4	70.9	54.4	48.8	36.2/15.8/31.4	84.0	82.0	68.8	66.4	95.7	94.1	87.4
Embedding model: `all-mpnet-base-v2`											
GPT-35-Turbo	62.6	33.4	40.9	35.5/15.4/30.3	78.0	83.0	54.5	58.0	89.2	78.9	83.1
GPT-4	69.6	55.4	50.0	35.9/15.5/31.2	83.0	81.0	64.9	63.0	95.4	93.9	84.1

Table 4: Comparisons of OpenAI models on CyberBench with two embedding models, `text-embedding-ada-002-2` and `all-mpnet-base-v2`, for retrieving five-shot similar examples besides the summarization task.

Model	**N.S.**	**Average**	**CyNER**	**APTNER**	**CyNews**	**SecMMLU**	**CyQuiz**	**MITRE**	**CVE**	**Web**	**Email**	**HTTP**
-	-	-	F1	F1	R-1/2/L	Acc.	Acc.	Acc.	Acc.	F1	F1	F1
Examples selected by similarity												
Llama-2-7B	1	28.4	15.2	18.2	0.3/0.3/0.3	48.0	47.0	24.2	46.9	22.0	25.4	36.7
	5	50.6	26.3	28.0	0.3/0.3/0.3	63.0	62.0	44.6	64.7	79.9	94.2	42.8
Llama-2-7B	1	26.8	7.3	9.3	25.2/9.6/21.6	57.0	52.0	18.3	35.7	20.1	8.2	41.1
-Chat	5	44.6	22.7	25.4	25.2/9.6/21.6	60.0	56.0	41.6	52.5	48.4	79.4	41.0
Examples selected randomly												
Llama-2-7B	1	29.2	10.0	9.4	0.3/0.3/0.3	39.0	51.0	1.3	42.6	57.2	53.4	27.5
	5	35.3	18.7	17.7	0.3/0.3/0.3	63.0	59.0	3.9	46.4	62.3	58.6	23.1
Llama-2-7B	1	32.6	8.8	8.8	25.2/9.6/21.6	52.0	53.0	1.2	34.0	51.5	49.9	48.5
-Chat	5	38.6	17.2	16.9	25.2/9.6/21.6	57.0	58.0	4.5	35.5	70.4	61.5	46.1

Table 5: Comparison of Llama-2-7B models with different few-shot examples on CyberBench. The zero-shot setting is always used for the summarization task. "N.S." is the number of shots.

Model	**Quant.**	**Average**	**CyNER**	**APTNER**	**CyNews**	**SecMMLU**	**CyQuiz**	**MITRE**	**CVE**	**Web**	**Email**	**HTTP**
-	-	-	F1	F1	R-1/2/L	Acc.	Acc.	Acc.	Acc.	F1	F1	F1
Llama-2-7B	4 Bit	50.3	27.2	27.8	1.0/0.4/0.8	60.0	57.0	43.1	63.0	82.2	93.6	48.2
	8 Bit	51.0	25.6	27.7	0.3/0.3/0.3	65.0	62.0	45.2	63.9	81.6	93.7	45.1
	Full	50.6	26.3	28.0	0.3/0.3/0.3	63.0	62.0	44.6	64.7	79.9	94.2	42.8
Llama-2-7B	4 Bit	45.4	22.7	27.2	24.1/9.3/20.8	60.0	50.0	40.7	52.7	60.2	85.7	36.2
-Chat	8 Bit	44.9	21.3	25.5	25.2/10.1/21.7	59.0	59.0	41.1	52.7	50.3	82.7	38.3
	Full	44.6	22.7	25.4	25.2/9.6/21.6	60.0	56.0	41.6	52.5	48.4	79.4	41.0

Table 6: Comparisons of Llama-2-7B models with different quantization precisions, including 4 bits, 8 bits, and full precision, on CyberBench, where `all-mpnet-base-v2` embedding model is used for retrieving five-shot similar examples besides the summarization task.