

# HackMentor: Fine-Tuning Large Language Models for Cybersecurity

Jie Zhang<sup>1,2</sup>, Hui Wen<sup>1,\*</sup>, Liting Deng<sup>1,2</sup>, Mingfeng Xin<sup>1,2</sup>, Zhi li<sup>1</sup>, Lun Li<sup>1</sup>, Hongsong Zhu<sup>1</sup>, Limin Sun<sup>1</sup>

<sup>1</sup>*Institute of Information Engineering, Chinese Academy of Sciences*

<sup>2</sup>*School of Cyber Security, University of Chinese Academy of Sciences*

BeiJing, China

{zhangjie, wenhui, dengliting, xinmingfeng, lizhi, lilun, zhuhongsong, sunlimin}@iie.ac.cn

**Abstract**—The democratization of artificial intelligence has made substantial progress by leveraging open-source large language models (LLMs), enabling researchers across domains to train customized models to meet their specific needs. Given the confidentiality and significance of cybersecurity, obtaining private and localized LLMs is imperative. However, general LLMs are not designed to cater specifically to this field, their general knowledge often falls short when addressing such specialized problems. In this paper, we categorize the domain instructions based on cybersecurity knowledge to guide the construction of high-quality instructions and conversations, ultimately enhancing the specialized capabilities of LLMs. The resulting fine-tuned LLMs, collectively termed HackMentor, are evaluated using WinRate, EloRating, and ZenoEval methods along with other popular LLMs. The experiments demonstrate that the proposed method yields significant performance improvements, surpassing the native LLMs by 10-25% when aligned with cybersecurity prompts. More, HackMentor exhibits comparable conversational quality to ChatGPT, while providing more concise and human-like responses. This study demonstrates the efficacy of HackMentor in augmenting LLMs for cybersecurity requirements, paving the way for localized LLMs that meet specialized needs without compromising general capabilities.

**Index Terms**—LLMs, cybersecurity, fine-tuning, instructions

## I. INTRODUCTION

The advancements of artificial intelligence, specifically large language models (LLMs) like GPT, have become widespread across various vertical fields, including cybersecurity. The success of ChatGPT demonstrates the remarkable ability of LLMs to follow instructions and generate human-like responses [1], [2]. The availability of open-source LLMs such as Llama and Bloom has facilitated research and progress in domain-specific language models, particularly in medical and legal fields [3]–[7]. Given the critical importance of cybersecurity for individuals and businesses, as well as the intricate nature and extensive knowledge in linguistic expression within the cybersecurity domain, it is imperative to develop LLMs that are specifically designed for cybersecurity.

Traditional research and products within the cybersecurity field primarily concentrate on detecting known threats and matching them using static features and rules [8]–[11]. However, these task-oriented approaches prove insufficient in dealing with rapidly evolving threats and hampers practitioners' comprehensive understanding of domain knowledge. Human

creativity and ability will always be imperative for defense. ChatGPT introduces a human-centric approach to utilizing LLMs, which can not only reason but also infer and detect unknown threats, employing natural language to facilitate more user-friendly interaction and support for security practitioners. Users no longer just play the role of providing input or receiving output, but gain the ability to analyze and resolve cybersecurity issues through interaction with LLMs. Therefore, the development of LLMs in the field presents numerous possibilities for AI-assisted cybersecurity applications.

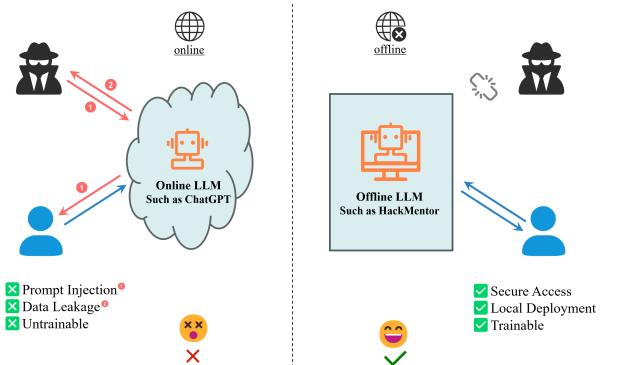


Fig. 1. Towards secure, locally deployable, and trainable domain LLMs.

Cybersecurity is a highly specialized field that demands precise and clear language expression. And existing LLMs are not designed to cater to this domain, their general domain knowledge often falls short when addressing such specialized fields. It is necessary to fine-tune LLMs for the domain to acquire the requisite expertise and language proficiency. As shown in Fig. 1, studies have proved that ChatGPT's online access poses security risks like prompt injection and data leakage [12], [13], considering the confidentiality and significance of cybersecurity, it is feasible to maintain data locally for training and utilizing LLMs. This approach allows organizations to fine-tune the models based on their own data and requirements, resulting in more accurate and customized results.

Fine-tuning LLMs with instructions has proven effective in enhancing their ability to follow instructions and improving overall performance [7], [14]–[16]. However, the scarcity of cybersecurity domain data required for training poses a signifi-

\*Corresponding author.

cant challenge, which stems from two primary reasons. Firstly, the data format in this domain does not align with the natural language data necessitated for fine-tuning LLMs. Previous datasets mainly comprised specialized domain data such as network traffic, logs, and binaries. Secondly, cybersecurity data is sensitive, fragmented, and imbalanced. Cybersecurity tasks encompass multiple interconnected domains, thereby complicating the construction of a comprehensive and exhaustive natural language dataset for this field.

This paper aims to fine-tune LLMs in the field of cybersecurity with constructed instructions and conversations. We first categorize instructions in the cybersecurity domain and utilize this to guide the construction of instructions and conversations that can be used to fine-tune LLMs. The Llama and Vicuna models are then fine-tuned using the low-rank adaptation (LoRA) method, resulting in a family of cybersecurity LLMs called HackMentor. We evaluate their performance on domain instruction following and conversational ability. As the LLMs demonstrate remarkable learning capabilities, fine-tuning them in specific fields can unleash their potential within professional domains. This study empowers cybersecurity professionals to effectively address tasks and challenges in the cybersecurity domain using HackMentor. The work has been made publicly available on <https://github.com/tmylla/HackMentor>.

The main contributions of this paper are as follows:

- To the best of our knowledge, our study is the first work to adapt LLMs to the cybersecurity domain.
- We propose a method using expert knowledge to systematically categorize domain instructions, which informs the construction of data for fine-tuning LLMs.
- We constructed 14k instructions and 30k conversations applicable to the field of cybersecurity.
- Our evaluation demonstrates HackMentor significantly outperforms original LLMs in specialized instruction following and conversation. The experiment shows increases ranging from 10-25%.

## II. RELATED WORK

### A. Fine-tuning LLMs

A certain parameter scale is a prerequisite for the strong power of LLMs. However, fine-tuning is still needed to enhance or unlock the performance and generalization to unseen tasks [17]. Currently, there are two methods for fine-tuning LLMs, namely instruction-based tuning [18], [19] and conversation-based tuning [20], [21].

Instruction-based tuning aims to guide LLMs in comprehending human instructions. Seif-Instruct [18] uses GPT-3 to generate 52k instructions based on 175 manually annotated seed instructions. Furthermore, GPT-3 is fine-tuned utilizing this data, resulting in a performance that is only 5% inferior to InstructGPT-001. Alpaca [22] performed instruction-based fine-tuning on Llama using 52k instructions generated by text-davinci-003. Guanaco [19] conducted instruction-based fine-tuning on Llama using 52k English instructions from Alpaca

and 534.5k multilingual instructions, resulting in excellent performance in a multilingual environment.

Conversation-based tuning is to train LLMs using multi-turn conversations so that LLMs can exhibit more natural and human-like chat abilities. Vicuna [21] is a pioneer of this research, which fine-tunes Llama using 70k user-shared conversations with ChatGPT collected from ShareGPT. It is one of the most advanced and versatile open instruction-following models. Similarly, Koala [23] has fine-tuned Llama utilizing 30k conversations, 169k instructions, and 273k preference-labeled data, producing responses that are comparable to ChatGPT in at least 50% of the cases. Phoenix [14] performed fine-tuning on BloomZ using 267k instructions and 189k conversations collected from over 40 languages, resulting in remarkable performance across multiple languages, including resource-limited ones.

### B. Domain-restricted LLMs Fine-tuning

Researchers from various disciplines have utilized domain-specific data to fine-tune LLMs in order to improve their accuracy and reliability in addressing specific tasks. In the medical field, ChatDoctor [6] has been developed as a healthcare consultation chatbot by fine-tuning Llama with authentic doctor-patient conversations. Also, it incorporates an external knowledge engine that leverages medical databases and Wikipedia to provide users with reliable and up-to-date information. HuaTuo [5], on the other hand, enhances the performance of Llama in Chinese medical question-answering by fine-tuning it with a dataset consisting of Chinese medical instructions created using medical knowledge graphs and ChatGPT. PMC-Llama [4] adopts a similar approach but focuses on incorporating medical knowledge from a vast dataset of 4.8 million biomedical academic papers, thereby augmenting its capabilities in the medical domain. Another notable example is Clinical Camel [20], which employs a dialogue-based knowledge encoding approach to transform MedQA data and publicly available clinical comments into conversations. This unique methodology, combined with fine-tuning Llama on ShareGPT, holds great promise as a valuable tool for healthcare applications.

In addition to the medical field, researchers in the legal and financial industries have also fine-tuned Llama to access domain-applicable LLM. Lawyer-Llama [3] collected the answer and question analysis from the Chinese National Unified Legal Professional Qualification Examination to fine-tune Llama with instruction, enabling the model to apply legal knowledge to specific scenarios. Cornucopia-Llama [24] gathers Chinese financial public data and crawled financial data to construct instructions and fine-tuned Llama to improve its performance in the financial field. As for LLM fine-tuning in the field of cybersecurity, there is currently no related research work.

### C. LLMs for Cybersecurity

Language models have a wide range of applications in the cybersecurity field. Researchers commonly employ pre-

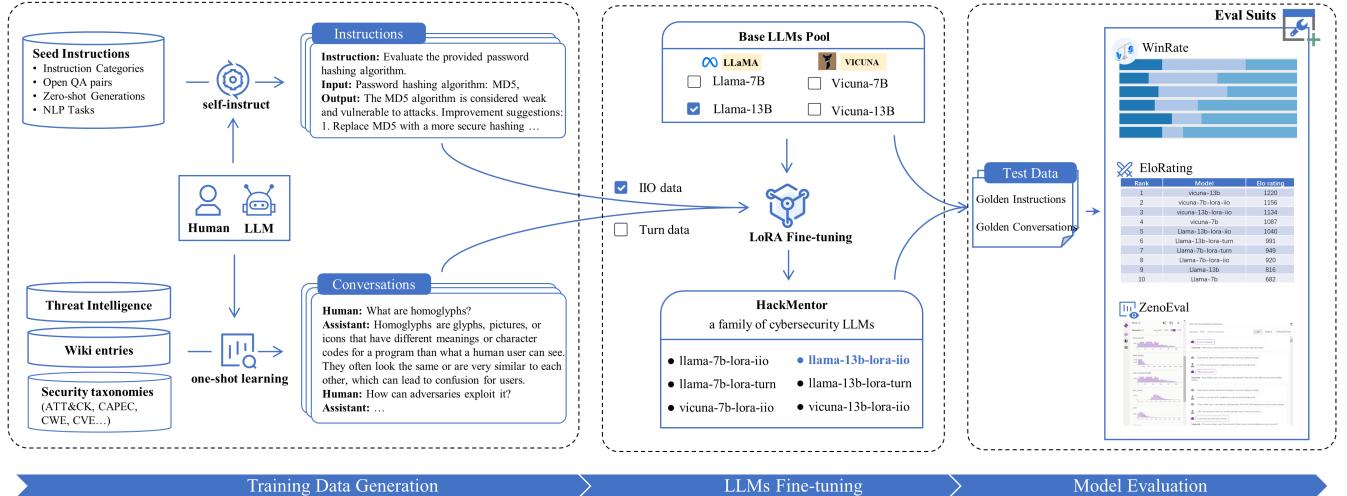


Fig. 2. The overview of HackMentor. HackMentor incorporates three components. The first one constructs the cybersecurity training data, including instructions (*iio data*) and conversations (*turn data*). The second part selects the training data and base model for LoRA fine-tuning and obtains a series of HackMentor, where the *llama-13b-lora-iio* (llama-13b fine-tuned with instructions) performs best. Lastly, the WinRate, EloRating, and ZenoEval are used to evaluate models' instruction following and human-computer conversation capabilities in the field of cybersecurity.

training models like BERT/RoBERTa and train them on cybersecurity texts to obtain pre-trained language models specifically for this domain, such as CySecBERT [25] and SecureBERT [26]. These models can serve as the foundational building blocks for downstream tasks, including classifying malicious software and answering cybersecurity-related questions. UTS [27], which is trained on the T5-base model, incorporates diverse security domain data such as malware reports, phishing website URLs, and programming codes. It conducts joint training on 10 cybersecurity tasks, encompassing both task transfer and domain transfer. As a result, UTS introduces a versatile, multi-nature, multi-task, and robust model capable of effectively handling new samples in the field of cybersecurity.

Currently, the application of LLMs in the field of cybersecurity is limited. There are only two commercially available LLMs for domain purposes, that is Microsoft's Security Copilot [28] and Google's Sec-PaLM [29]. However, these models are still in the research and development phase and have not yet reached maturity. Their main objective is to enhance existing detection efforts. Security Copilot utilizes LLM and security-specific models to deploy skills and queries, assisting security personnel in faster and more accurate detection and response to security threats. Sec-PaLM, available via Google Cloud, employs AI to analyze and interpret the behavior of potential malicious scripts. Despite their capabilities, these models have significant limitations when it comes to private deployment and customization for personalized tasks. Consequently, there is an urgent need for research in the cybersecurity domain to develop locally fine-tuned LLMs.

### III. METHODOLOGY

This section presents an overview of HackMentor, the LLMs fine-tuned in the cybersecurity domain. As shown in Fig. 2, the framework consists of three main components: training data

generation, LLMs fine-tuning, and model evaluation. First, we construct specific instructions and conversations tailored for LLM fine-tuning in cybersecurity. Next, we perform LoRA fine-tuning on the Llama and Vicuna models of 7B and 13B using instructions and conversations, respectively. Lastly, we conduct a comprehensive evaluation of LLMs using the suits of the WinRate, EloRating, and ZenoEval methods.

#### A. Training Data Generation

To fine-tune LLMs for specific domains, such as cybersecurity, high-quality training data is required. Previous studies like Llama [7] and QLoRA [19] have demonstrated that data quality is more important for developing and enhancing the capabilities of LLMs compared to data quantity. However, currently, there is a lack of training data available for fine-tuning LLMs in the field of cybersecurity. To address this issue, we first compiled and constructed instructional and conversational data for the cybersecurity domain, guided by richness, comprehensiveness, and security considerations.

1) *Cybersecurity Instructions*: An instructional data typically consists of instruction, input, and output. Instruction is the task descriptions for LLMs, input is the optional context for the instruction task, and output is the answers generated by large language models. In this paper, the construction of the instructions is divided into three parts: defining security instruction categories, creating seed instructions, and generating a comprehensive instructional dataset. Firstly, we classify and organize task instructions related to cybersecurity. Subsequently, we construct 144 seed instructions for the domain. Finally, we utilize the self-instruct method, relying on gpt-3.5-turbo, to generate a dataset comprising 14k instructions.

##### a) Definition of Cybersecurity Instruction Categories:

Research has shown that LLMs outperform humans as data annotators [30], [31]. Within this context, we combined human expertise with state-of-the-art GPT-4 to categorize instruction

TABLE I  
INSTRUCTION CATEGORY

Instruction Category	Description	Subcategory Count
Concept Explanation	Explain various concepts in cybersecurity, such as DDoS attacks, encryption algorithms, etc.	15
Case Study	Provide a cybersecurity case for the respondent to analyze the causes, impacts, and solutions.	17
Technical Solution	Present a cybersecurity requirement or application scenario and ask for a technical implementation and solution.	14
Practical Exercise	Provide a simulated network environment for the respondent to engage in attack and defense exercises.	7
Security Management	Propose cybersecurity management requirements and ask the respondent to formulate management strategies or plans, including security training, awareness, etc.	14
Policy and Regulations	Require explanations of laws and regulations related to network security, including privacy protection, data security, etc.	10
Scenario Assessment	Provide a virtual cybersecurity scenario for the respondent to make judgments and decisions.	11
Industry Applications	Explore the application of cybersecurity in a specific industry, such as finance, healthcare, retail, etc.	9

types in the field. Firstly, we meticulously curated 8 major categories of domain-related instruction types, as displayed in Table I. Then, considering the variations in diversity, content completeness, and the distinct forms of instructions, inputs, and outputs within each category, we employed LLMs to generate 200 sub-instructions for each category. Through a meticulous process of manual review and selection, we derived a final set of 97 sub-categories of security instruction types specific to the cybersecurity domain.

*b) Generation of Seed Instructions for Cybersecurity:*

The construction of seed instructions requires manual curation to guarantee quality, usability, and effectiveness. Therefore, we prioritized diversity, scalability, effectiveness, and harmlessness when generating seed instructions. Our approach also encompassed a thorough consideration of instructions, inputs, and outputs throughout the construction process, with corresponding measures in place. Regarding instruction and input, we aimed to encompass various knowledge aspects pertinent to the cybersecurity domain, including theoretical knowledge, practical solutions, tool skills, and academic tasks, among other diverse topics. In terms of outputs, we ensured alignment with the instruction inputs while diversifying the output styles, such as utilizing text, tables, and topological structures. Furthermore, we implemented robust filtering mechanisms to exclude data that could potentially facilitate malicious behavior, ensuring the safety and harmlessness of the seed instructions.

The composition of the 144 seed instructions developed within the scope of this study, focused on the cybersecurity domain, is outlined as follows:

*Open QA Data* To start, we gathered over 200 QA pairs from the Internet. Then, we carefully selected 15 pairs as seed instructions, taking into account the types of answers and the QA styles.

*GPT-4 Zero-shot Generation* Drawing from Alpaca's prompt, we customized the seed prompt to suit the specific requirements of the security domain, as elaborated in Appendix A. Subsequently, GPT-4 is employed to generate one hundred instances of instruction, input, and output (IIO) data based on tailored prompts. Through meticulous manual screening, considering factors such as instruction repetition and expression richness, we derived our final set of 19 seed instructions.

*Cybersecurity Instruction Categories* According to the aforementioned 8 major categories and 97 sub-categories, we required LLMs to provide five question-answer pairs. To ensure the correctness, richness, and reasonableness of the answers, we conducted manual scoring on a scale of 1 to 5. Instructions with a score of 5 were selected as seed instructions, resulting in a total of 97 seed instructions.

*Cybersecurity NLP Tasks* UTS [27] categorized cybersecurity NLP tasks into 4 major categories: classification, named entity recognition, event detection, and regression. These categories are further divided into 13 sub-categories, which include malware text detection, malware entity relation identification, hacker's threat detection, spam message detection, phishing URL detection, vulnerable tweet detection, event argument role identification, information security entities extraction, cybersecurity entity detection, computer programming entity extraction, event nuggets extraction, detect arguments of event from the sentence, vulnerability impact score estimation. We collected one instruction for each task, resulting in a total of 13 seed instructions.

*c) Construction of Cybersecurity Instructions:* Based on the seed instructions in the field of cybersecurity, we employed the self-instruct method to construct the cyber-security instruction dataset. Notably, we utilized the gpt-3.5-turbo model for this purpose. During the generation of cyber-instructions, we calculated the Rouge-L scores between the newly generated instructions and the existing instructions, with a threshold value of 0.6. To ensure the diversity of the instructions, we only retained the generated data with lower Rouge-L scores. It is worth noting that we excluded the seed instructions from the instruction dataset, thus we preserved the seed instructions as test instructions, which will be discussed in Section IV. Furthermore, we visualized the distribution of verbs and nouns in the instructions and inputs within the cybersecurity instruction dataset. We plot the top 20 most common root verbs and their top 4 direct noun objects in Appendix B.

*2) Cybersecurity Conversations:* The field of cybersecurity faces challenges in terms of standardized conversation datasets, as the available long conversations from QA communities primarily consist of open discussions involving multiple individuals, rather than interactive conversation. And cybersecurity conversation is scarce and difficult to extract in the

ShareGPT dataset. Drawing inspiration from LlamaIndex, we sought to address this issue by incorporating an additional knowledge base specific to the cybersecurity domain. Moreover, we leveraged ChatGPT to generate conversations within this field.

The cybersecurity knowledge base primarily consists of unstructured text, including threat intelligence from 2008 to 2021, Wikipedia entries on cybersecurity topics, and descriptions of cybersecurity taxonomies such as *ATT&CK*, *CAPEC\_AP*, *CAPEC\_COA*, *CWE*, and *CVE*. Generating conversations based on this knowledge base ensures data authenticity. To ensure data rationality, we adhere to four principles when rewriting conversation generation prompt: cold start, domain dependency, contextual relevance, and logical hierarchy. In other words, when engaging in conversations related to the cybersecurity domain, the first sentence should not contain ambiguous content like "Do you know about this vulnerability?" Instead, it should specifically mention the ID or name of the vulnerability. The number of conversation turns and the logical correlation between turns depend on the text content, considering the context, coreference resolution, and chronological order.

One-shot learning is also used to enhance the LLM's understanding of prompts, we provide an example to facilitate the few-shot learning of ChatGPT, thus generating the desired conversations. For cybersecurity conversation generation prompts, please refer to Appendix A.

#### B. LLMs Fine-Tuning with LoRA

*Base model* We have chosen Llama and Vicuna as the anchor LLMs, after which we conduct domain adaptation fine-tuning by utilizing cybersecurity instructions and conversations. Regarding Llama, we select the 7B and 13B models, both trained on 1T tokens by Meta. Vicuna is an open-source chatbot trained by fine-tuning Llama on user-shared conversations collected from ShareGPT. Similarly, we use the 7B and 13B models for fine-tuning.

*LoRA Training* We employ the LoRA to perform fine-tuning of LLMs. The principle of LoRA [32] involves freezing the weights of the original model and exclusively training the additional network layers by incorporating them, which offers several advantages, including rapid training speed, minimal computational demands, and lightweight training weights. Ablation experiments demonstrate that fine-tuning  $W_q$  and  $W_v$  yields comparable performance to fine-tuning all  $W_q$ ,  $W_k$ ,  $W_v$ , and  $W_o$ . Consequently, in our experiments, we only adjust  $W_q$  and  $W_v$ . We conduct separate LoRA fine-tuning on Llama and Vicuna using cybersecurity instructions and conversations. The trained LoRA weights can be flexibly integrated with the original weights to cater to specific inference requirements.

*Training Parameters* The models are implemented in PyTorch using the Huggingface Transformers package. We conduct the training on an A100 80G GPU. The general settings for the training are as follows: a maximum length of 1024, optimization using the AdamW optimizer, a batch size of 32, and a weight decay of 0. For the 7B and 13B LLMs (both

Llama and Vicuna), the learning rates are set to 2e-5 and 1e-5, respectively. The instructions are trained for 8 epochs, whereas the conversations are trained for 3 epochs. Regarding the LoRA parameters, the specific settings are as follows: the LoRA attention dimension of 8, the names of the modules to apply LoRA to the 'q\_proj' and 'v\_proj' modules, the alpha parameter of 16 for LoRA scaling, and the dropout probability of 0.05 for LoRA layers.

As mentioned above, we use the Llama and Vicuna as the base models and fine-tune them with LoRA approach on the constructed instructions and conversations to obtain a series of cybersecurity LLMs, namely HackMentor. Causing different combinations of training data and base models, each individual LLM in HackMentor is named as follows:  $\{\text{BaseModel}\}-\{\text{ModelSize}\}-\{\text{Fine-tuning method}\}-\{\text{Data type}\}$ , where BaseModel can be Llama or Vicuna, ModelSize can be 7b or 13b, Fine-tuning method is by default Lora, and DataType includes iio and turn, iio means instruction, input, and output data, that is, instructions, while turn represents conversations. For example, llama-13b-lora-iio indicates the fine-tuning of the 13B Llama model using instructions with LoRA. In the end, we fine-tune a total of six domain LLMs in HackMentor: llama-7b-lora-iio, llama-7b-lora-turn, llama-13b-lora-iio, llama-13b-lora-turn, vicuna-7b-lora-iio, and vicuna-13b-lora-iio. Among these models, llama-13b-lora-iio performs the best and is the top choice of HackMentor. The reason Vicuna is not fine-tuned with security conversations is primarily due to the results on Llama indicating that instruction is a better choice. More analysis can be found in Section IV.

#### C. Model Evaluation

We use the evaluation suit including WinRate, EloRating, and ZenoEval to assess the performance of the fine-tuned models on cybersecurity knowledge and chat interaction with humans. WinRate and EloRating are used to compare models' proficiency in cybersecurity knowledge. WinRate evaluates the response quality by comparing the custom model with ChatGPT, resulting in three possible scenarios: custom win, tie, or ChatGPT win. A higher proportion of custom wins and ties indicates better performance by the custom model. EloRating, on the other hand, is an enhanced version of WinRate. It involves placing models in a pool and randomly selecting two models to respond to the same question, which provides a record of 'win rates' between pairs of models, which we can then fit corresponding Elo scores [33].

During the pairwise comparison of models, a total of 6,525 combinations of results will be generated. This is because there are 10 models involved in the comparison, and  $145 \times C_{10}^2 = 6525$ . Previous research has demonstrated that LLM-as-judge is a scalable and explainable way to approximate human preferences, which are otherwise very expensive to obtain. Taking into consideration both the cost and quality of models, we have chosen to compare the results of each model pair using Claude. The design of the comparison prompt can be found in Appendix A. Claude evaluates the response results of two anonymous models based

on several factors, including exact match, content relevance, coherence and logic, accuracy, and response quality. Firstly, Claude checks if 'model-A-output' and 'model-B-output' are exactly the same, if so, the result is a "tie". If the outputs differ, factors such as content relevance, coherence and logic, accuracy, correctness, and overall content quality are taken into consideration to determine which model's response is superior. If it is still not possible to determine which model is better after evaluating these factors, the result will be a "tie". See Appendix C for more details.

ZenoEval focuses on enhancing the chat experience and evaluating the fluency, coherence, and understandability of the content generated by various models in human-machine conversations. Model responses are assessed using state-of-the-art text evaluation metrics from Critique [34]. The selected metrics in this paper include:

- ChrF: Measures the overlap of character strings.
- BERTScore: Measures overlap of embeddings between the two utterances.
- Length ratio: Measures the length of the output divided by the length of the gold-standard human response.
- Coherence: Predicts how coherent the outputs are with the previous chat turn.
- Naturalness: Judge whether the response is like something a person would naturally say.
- Understandability: Judge whether the response is understandable.

Moreover, ZenoEval offers an interactive visualization interface that allows users to examine the responses of various models and analyze examples with lower scores. This interface provides valuable insights for understanding the performance of the models. Additionally, it enables the analysis of the correlation between evaluation metrics and conversation position or the length of human gold responses. This correlation analysis aids in identifying patterns and trends, supporting the improvement and enhancement of the models.

#### IV. EXPERIMENT RESULTS

This section evaluates the performance of HackMentor and the baseline on cybersecurity instruction and conversation test data. The findings demonstrate the proposed model improves cybersecurity instruction following by 10-25% compared to native LLMs, but with the added advantage of supporting local deployment and application to ensure data privacy and security. Additionally, our model exhibits comparable chat quality to ChatGPT, while providing more concise responses that align with typical human conversational habits.

##### A. Test Dataset in Cybersecurity

**Instruction Test Data** In our experiment, seed instructions are not used to fine-tune the LLMs, and the ROUGE score between the seed instructions and training instructions is less than 0.6, making it suitable for test data.

**Conversation Test Data** A total of 70 conversations were manually collected for testing, counting 316 rounds and the average length of each conversation was 9. We selected 10

longer conversations from discussions about cybersecurity topics on Reddit. We designed prompts for GPT-4 to generate 25 conversations, which are based on professional scenarios in the field. The prompts for scenario-based conversations can be found in Appendix A. We sampled 15 unused conversations from our train dataset. Additionally, we ensured the freshness of the data by generating 10 new conversations based on blog content published after 2023. Furthermore, we manually designed 10 conversations in various cybersecurity scenarios.

##### B. Baseline

The evaluation of the instruction and conversation test data involved a comparative analysis between five baseline models and the HackMentor LLMs in the cybersecurity domain, namely ChatGPT, Llama 7b & 13b, Vicuna 7b & 13b. *ChatGPT* is one of the most widely used and powerful LLMs available today, which serves as a benchmark for a majority of LLM research. *Llama* stands as a highly popular open-source LLM among researchers, with the 7B and 13B models as the baselines for instruction and conversation fine-tuning. *Vicuna* is one of the most advanced and popular open-domain chat models, it also has 7B and 13B models available, which are all used for instruction fine-tuning.

##### C. Result Analysis

1) *WinRate, Comparison between HackMentor and ChatGPT in Cybersecurity Instructions:* We compared the custom model with ChatGPT on cybersecurity instructions. Our HackMentor shows significant performance improvement compared to the baseline model, achieving comparable results to ChatGPT. Additionally, unlike the online access method of ChatGPT, HackMentor supports local deployment of applications, providing a more secure, private, and reliable LLM service.

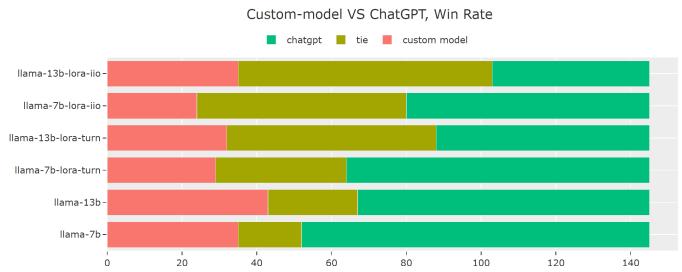


Fig. 3. WinRate between Custom-model and ChatGPT. The left side means custom-model wins, the middle means a tie and the right side means ChatGPT wins. The better custom-model, the fewer wins for ChatGPT.

We evaluated the responses of the custom model and ChatGPT to each instruction, categorizing the outputs into three categories: the custom model win, tie, and ChatGPT win. The WinRate results of six custom models and ChatGPT are shown in Fig. 3. *llama-7b/13b* represents the original *llama* model, *llama-7b/13b-lora-turn* is HackMentor that fine-tuning *llama* with LoRA using domain conversations, and *llama-7b/13b-lora-iio* is HackMentor that fine-tuning *llama* with LoRA using domain instructions.

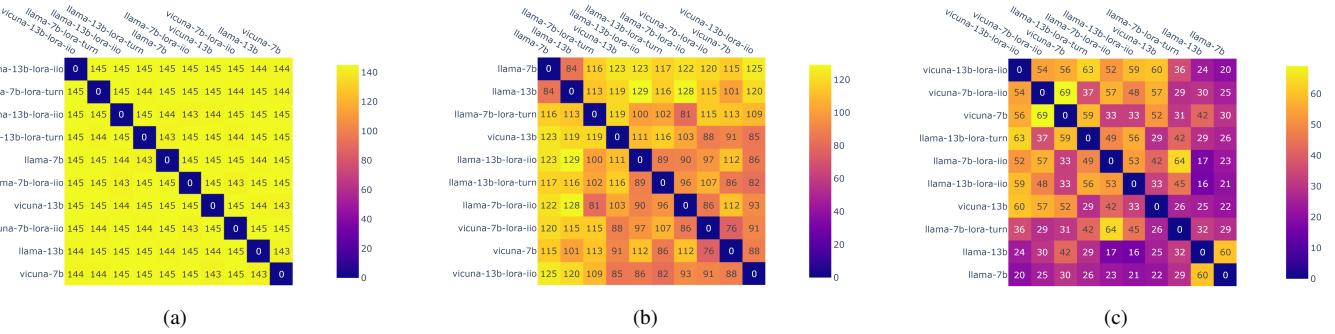


Fig. 4. (a): Battle count of each combination of models. (b): Battle count of each combination of models (without ties). (c): Tie count of each combination of models.

TABLE II  
ELO RATING RANK

Rank	Model	Elo rating
1	vicuna-13b	1220
2	vicuna-7b-lora-iio	1156
3	vicuna-13b-lora-iio	1134
4	vicuna-7b	1087
5	llama-13b-lora-iio	1040
6	llama-13b-lora-turn	991
7	llama-7b-lora-turn	949
8	llama-7b-lora-iio	920
9	llama-13b	816
10	llama-7b	682

Fine-tuned Llama models have shown performance improvements of 10% to 25% in specific knowledge question-answering. Compared to conversations, instructions possess better capability in injecting domain knowledge into LLM. This speculation is based on the fact that this stage was performed using instruction test data. Other studies have presented evidence that instruction fine-tuning is a superior approach to conversation fine-tuning, with Phoenix achieving a relative improvement of 5%-6% with instruction fine-tuning compared to conversation fine-tuning. It is important to note that the fine-tuned Llama model has slightly fewer absolute victories than the original Llama model but achieves more tie results with ChatGPT. This suggests that the presence of an "alignment tax" [35] during the fine-tuning process, which aligns with ChatGPT, partially impacts the original performance of the Llama model.

In addition, HackMentor achieves comparable results with ChatGPT, and its smaller size and offline capability facilitate local deployment. Among the 144 test instructions, llama-13b-lora-iio wins 35 times, ChatGPT wins 42 times, and there are 68 ties. With a slight margin of weakening compared to ChatGPT (7/144), suggesting the proposed model is highly competitive in following domain instructions. Moreover, the online access of ChatGPT poses security risks like prompt injection and data leakage. In contrast, HackMentor supports offline deployment, allowing further training with user-specific data, which is indispensable for conducting sensitive and critical security research.

2) *EloRating, HackMentor vs. Base-LLMs in Chatbot Arena:* The HackMentor series models, along with Llama and Vicuna models, were included in the model pool. We employed the ChatBot Arena method to compare and rank all models based on the comparison results.

We use Claude to evaluate the response of two anonymous models to the same test data, determining the model with a better response or a tie result. We conduct A/B tests on 10 models respectively, resulting in a total of  $C_{10}^2 * 145 = 6525$  pairwise comparisons. Fig. 4 shows the statistics of battle count for each model pair.

Based on the statistics of each model combination, we calculated the elo ratings for different models, as shown in Table II. Firstly, fine-tuning Llama with either conversations or instructions consistently resulted in improved performance over the original models. And, the larger base model (13b > 7b) achieved higher scores. Notably, the use of instructions had a more significant impact on improving performance compared to conversations. As for why Vicuna usually demonstrates better performance, both in the general domain and the cybersecurity domain (discussed in this paper), the reason can be attributed to the training data used. It is well known that the quality of data plays a crucial role in determining the quality of both pre-trained and fine-tuned models. The incorporation of abundant and authentic human-to-ChatGPT conversations from shareGPT contributes to the exceptional capabilities of Vicuna. Moreover, when fine-tuning Vicuna with instructions, it was observed that while the fine-tuned vicuna-7b showed improved performance in the field of cybersecurity, the fine-tuned vicuna-13b weakened the ability of the original model to follow instructions. This finding suggests that the enhanced instruction-following ability observed in the general domain can be generalized to specific capabilities. Further research is needed to explore how to achieve and surpass the capabilities of Vicuna in cybersecurity.

3) *ZenoEval, Evaluation of LLMs' Conversation Capability:* We conducted evaluations on a total of 12 models using conversation test data, including text-davinci-003, gpt-3.5-turbo, llama-7b/13b, vicuna-7b/13b, as well as HackMentor that models fine-tuned for cybersecurity. The evaluation metrics contain chrF, BERTScore, length ratio, coherence,

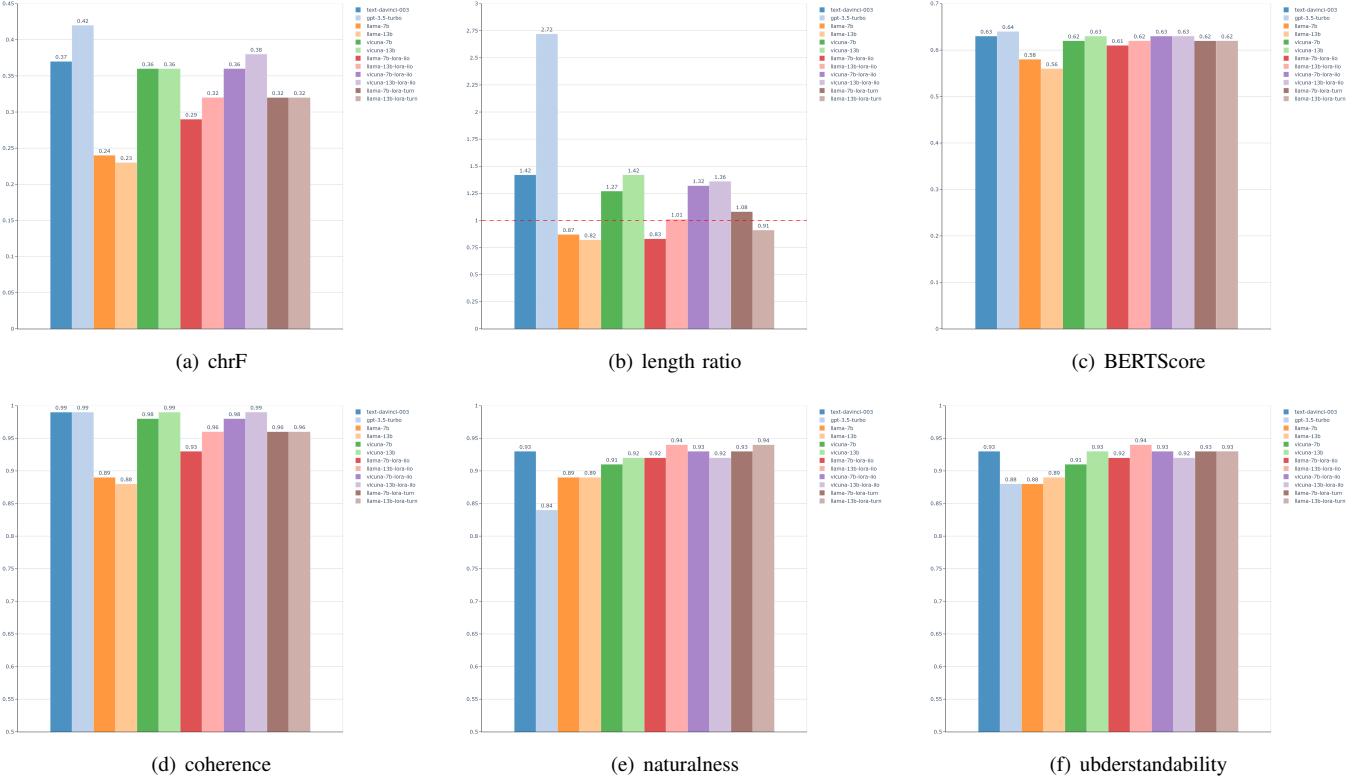


Fig. 5. Comparison of overall evaluation metrics. The performance of 12 models across 6 metrics is measured using the cybersecurity conversation test data. In general, a larger metric means a better model, with the exception of length\_ratio, which measures the ratio of the generated output length to the golden human response length. A value approaching 1, the red dash in (b), suggests an output that closely matches the golden response.

naturalness, and understandability.

During the evaluation, we performed experiments by setting different temperatures and context windows. The temperature is used to control the randomness of the generated text by LLM, the lower temperature generates more deterministic and consistent text, and the higher temperature generates more diverse and innovative text. The context window refers to the number of previous conversation turns given to LLMs when generating a new response. In our experiment, temperature can be 0.3, 0.6, or 0.9, and the context window can be 1, 2, 3, or 4. We conducted experiments on all models with different parameter combinations, and the experimental results of some parameter combinations can be found in Appendix D. The results indicated that the models performed best under the combination of temperature 0.3 and a context window of 4. We then selected this parameter combination for further analysis.

a) *Comparison of Overall Evaluation Metrics:* The overall evaluation of the 6 metrics, including chrF, BERTScore, length ratio, coherence, naturalness, and understandability, is shown in Fig. 5.

For the chrF and length-ratio metrics, as shown in Fig. 5(a) & 5(b), there are significant differences in performance among each model. In chrF metric, gpt-3.5-turbo achieves the highest score, with vicuna-13b-lora-iio closely following. Both Llama and Vicuna models show improvements in chrF after fine-tuning, especially the Llama models. The 7b/13b models fine-

tuned with instructions and conversations demonstrate chrF improvements of up to 5%/9% and 8%/9%, respectively. As for the length ratio metric, the outputs of gpt-3.5-turbo are generally longer in comparison to other models. The responses generated by text-davinci-003 and vicuna are slightly longer than the ideal length, while those from Llama tend to be shorter. In contrast, the length of outputs from the fine-tuned models is comparable to the ideal length. For example, the length ratio of llama-13b-lora-iio compared to the ideal length is 1.01.

For the bert-score and coherence metrics, as shown in Fig. 5(c) & 5(d), models except llama-7b and llama-13b demonstrate similar performance, indicating that HackMentor outperforms native models in cybersecurity conversations. Additionally, it can be observed that the llama model fine-tuned with conversations is more helpful in developing LLM conversational skills than instructions.

For the naturalness and understandability metrics, as shown in Fig. 5(e) & 5(f), each model performs well, indicating that LLMs are capable of accomplishing tasks related to natural language understanding and generation. Notably, llama-13b-lora-iio and llama-13b-lora-turn achieve the highest scores, while gpt-3.5-turbo obtains the lowest score. This can be attributed to two primary factors: on one hand, the model fine-tuned with cybersecurity data is more sensitive and effective in the specific domain; on the other hand, gpt-3.5-turbo tends

to generate longer responses, which affects its naturalness and understandability.

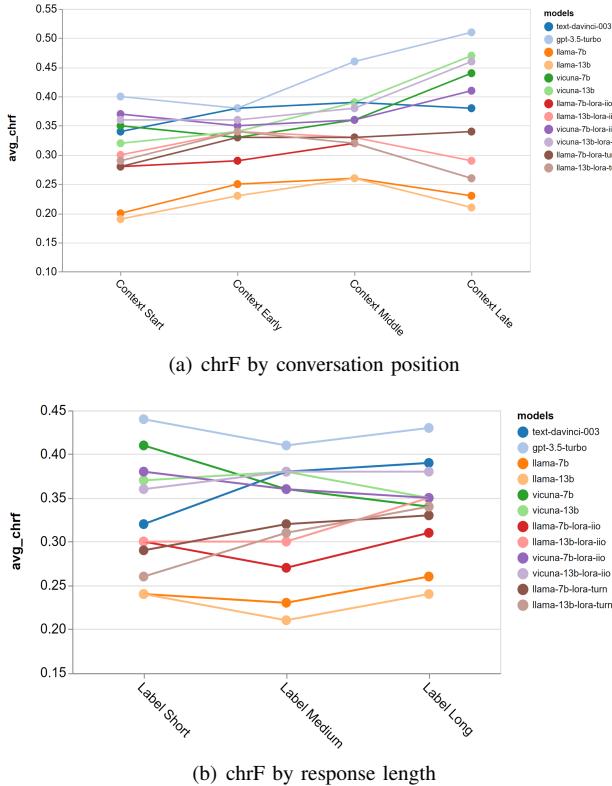


Fig. 6. Correlation analysis of chrF metric with conversation position or response length respectively.

*b) How do Conversation Position or Response Length Affect LLM Performance:* Different from the analysis of single-response instructions, the conversation position of turns and the inconsistency in the length of each turn pose greater challenges to LLMs. To investigate the performance of LLMs in cybersecurity conversations, we divided the test data based on conversation positions and the length of human responses. We compared the model's responses under different data characteristics. We divided conversation positions into 4 groups: start (cold start with no prior conversation), early (after 1-3 rounds of conversation), middle (after 4-7 rounds of conversation), and late (beyond 7 rounds of conversation). Response lengths were categorized into 3 groups: short ( $\leq 35$  characters), medium (36-70 characters), and long ( $\leq 71$  characters).

Fig. 6 shows the variations in chrF metric among different models, considering conversation position and response length. In Fig. 6(a), the chrF values vary within a range of 0.05 with respect to different response lengths, indicating that LLMs are well adaptable to responses of different lengths. Fig. 6(b) demonstrates that LLMs perform the worst at the beginning of conversations but achieve the highest performance in context early, and throughout the middle and late stages, their performance remains consistent. This is attributable to the context window being set to 4. In context start, models lack prior information, resulting in poor performance. In context early,

the window of 4 allows models to incorporate all context from the previous 4 turns, leading to improved performance. In context middle or late, models are limited to the previous 4 turns and are unable to access earlier information, which results in inferior performance compared to the early stage. Besides, regardless of conversation positions or response lengths, models fine-tuned show better performance compared to the original models, consistent with the previous findings.

We also compared the relationship between the length ratio and response length, as shown in Fig. 7. When the responses are accurate, shorter replies are associated with higher readability and comprehensibility. However, it can be observed that LLMs tend to generate longer responses, resulting in a decrease in the length ratio as the response length increases. In short response instances, gpt-3.5-turbo has the longest response, followed by text-davinci-003 and Vicuna models, while the Llama models have the shortest response. In medium and long response instances, all models except gpt-3.5-turbo can output responses of appropriate length. Additionally, our HackMentor models demonstrate robustness across various response lengths within a relatively stable range when compared to the standard length. This implies the proposed model has aligned its conversational features with the cybersecurity domain during the fine-tuning process, making it more suitable for interaction with humans compared to other LLMs.

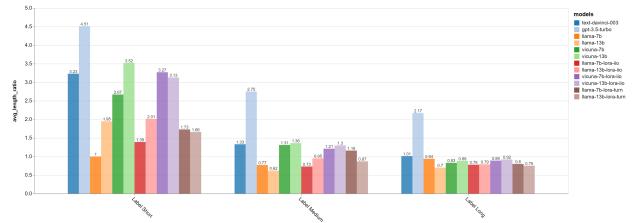


Fig. 7. The relationship between the length ratio metric and response length.

## V. CONCLUSION

We propose an innovative classification and definition of instructions in the cybersecurity field, aiming to guide the construction of instructions and conversations for fine-tuning LLMs. We utilize constructed domain instructions and conversations to fine-tune LLMs, resulting in cybersecurity LLMs called HackMentor. HackMentor can be deployed and applied locally to protect users' data privacy and security. Unlike previous task-driven AI assistance, the natural language interaction between users and HackMentor ensures that humans regain the dominant position in security decision-making. AI acts as a copilot and guides users when necessary, akin to a mentor. The release of the LoRA weights in this study is intended to facilitate cybersecurity practitioners in addressing tasks and challenges with the assistance of LLMs.

This paper acknowledges that the available training data in the field of cybersecurity is still inadequate and incomplete. We will conduct further research to augment the dataset and incorporate new data for all-parameters fine-tuning LLMs,

aiming to develop more competitive and practical LLMs in the field of cybersecurity.

#### ETHICS STATEMENT

The rapid development and adoption of large language models have been accompanied by a rigorous conversation about the ethics, uses, and limitations of these models. Cybersecurity is a highly sensitive field that demands the assurance of data privacy, security, integrity, and availability. To offer LLM services to the domain community, we have established a meticulous training and testing evaluation process. We incorporate various measures, including harmless prompts, keyword filtering, and manual verification, throughout the entire process to ensure the security of the data. However, we must acknowledge that we cannot guarantee absolute security for the models, and there is a possibility of errors in the outputs of HackMentor. If you require practical applications, it is advisable to seek professional assistance.

#### REFERENCES

- [1] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, and others, “A survey of large language models,” *arXiv preprint arXiv:2303.18223*, 2023.
- [2] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, and others, “Emergent abilities of large language models,” *arXiv preprint arXiv:2206.07682*, 2022.
- [3] Q. Huang, M. Tao, Z. An, C. Zhang, C. Jiang, Z. Chen, Z. Wu, and Y. Feng, “Lawyer LLaMA Technical Report,” *arXiv preprint arXiv:2305.15062*, 2023.
- [4] C. Wu, X. Zhang, Y. Zhang, Y. Wang, and W. Xie, “Pmc-llama: Further finetuning llama on medical papers,” *arXiv preprint arXiv:2304.14454*, 2023.
- [5] H. Wang, C. Liu, N. Xi, Z. Qiang, S. Zhao, B. Qin, and T. Liu, “Huatuо: Tuning llama model with chinese medical knowledge,” *arXiv preprint arXiv:2304.06975*, 2023.
- [6] L. Yunxiang, L. Zihan, Z. Kai, D. Ruilong, and Z. You, “Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge,” *arXiv preprint arXiv:2303.14070*, 2023.
- [7] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, and others, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [8] M. R. Rahman, R. M. Hezaveh, and L. Williams, “What are the attackers doing now? Automating cyberthreat intelligence extraction from text on pace with the changing threat landscape: A survey,” *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–36, 2023, publisher: ACM New York, NY.
- [9] A. Alsaheel, Y. Nan, S. Ma, L. Yu, G. Walkup, Z. B. Celik, X. Zhang, and D. Xu, “[ATLAS]: A sequence-based learning approach for attack investigation,” in *30th USENIX security symposium (USENIX security 21)*, 2021, pp. 3005–3022.
- [10] F. Abazari, E. Branca, N. Ridley, N. Stakhanova, and M. Dallapreda, “Dataset characteristics for reliable code authorship attribution,” *IEEE Transactions on Dependable and Secure Computing*, 2021, publisher: IEEE.
- [11] Z. Jadidi and Y. Lu, “A threat hunting framework for industrial control systems,” *IEEE Access*, vol. 9, pp. 164 118–164 130, 2021, publisher: IEEE.
- [12] R. Khouri, A. R. Avila, J. Brunelle, and B. M. Camara, “How Secure is Code Generated by ChatGPT?” *arXiv preprint arXiv:2304.09655*, 2023.
- [13] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, “More than you’ve asked for: A Comprehensive Analysis of Novel Prompt Injection Threats to Application-Integrated Large Language Models,” *arXiv preprint arXiv:2302.12173*, 2023.
- [14] Z. Chen, F. Jiang, J. Chen, T. Wang, F. Yu, G. Chen, H. Zhang, J. Liang, C. Zhang, Z. Zhang, and others, “Phoenix: Democratizing chatgpt across languages,” *arXiv preprint arXiv:2304.10453*, 2023.
- [15] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, and others, “Training language models to follow instructions with human feedback,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730–27744, 2022.
- [16] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, and others, “Scaling instruction-finetuned language models,” *arXiv preprint arXiv:2210.11416*, 2022.
- [17] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, “Scaling laws for neural language models,” *arXiv preprint arXiv:2001.08361*, 2020.
- [18] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, “Self-instruct: Aligning language model with self generated instructions,” *arXiv preprint arXiv:2212.10560*, 2022.
- [19] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “Qlora: Efficient finetuning of quantized llms,” *arXiv preprint arXiv:2305.14314*, 2023.
- [20] A. Toma, P. R. Lawler, J. Ba, R. G. Krishnan, B. B. Rubin, and B. Wang, “Clinical Camel: An Open-Source Expert-Level Medical Language Model with Dialogue-Based Knowledge Encoding,” *arXiv preprint arXiv:2305.12031*, 2023.
- [21] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, “Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%+ ChatGPT Quality.” Mar. 2023. [Online]. Available: <https://lmsys.org/blog/2023-03-30-vicuna/>
- [22] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, “Stanford alpaca: An instruction-following llama model,” 2023. [Online]. Available: [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca)
- [23] X. Geng, “Easylm: A simple and scalable training framework for large language models,” Mar. 2023. [Online]. Available: <https://github.com/young-geng/EasyLM>
- [24] Y. Yu, “Cornucopia-LLaMA-Fin-Chinese,” 2023. [Online]. Available: <https://github.com/jerry1993-tech/Cornucopia-LLaMA-Fin-Chinese>
- [25] M. Bayer, P. Kuehn, R. Shanehsaz, and C. Reuter, “CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain,” *arXiv preprint arXiv:2212.02974*, 2022.
- [26] E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer, “SecureBERT: A Domain-Specific Language Model for Cybersecurity,” in *International Conference on Security and Privacy in Communication Systems*. Springer, 2022, pp. 39–56.
- [27] K. K. Pal, K. Kashihara, U. Anantheswaran, K. C. Kuznia, S. Jagtap, and C. Baral, “Exploring the Limits of Transfer Learning with Unified Model in the Cybersecurity Domain,” *arXiv preprint arXiv:2302.10346*, 2023.
- [28] V. Jakkal, “Introducing Microsoft Security Copilot: Empowering defenders at the speed of AI,” Mar. 2023. [Online]. Available: <https://blogs.microsoft.com/blog/2023/03/28/introducing-microsoft-security-copilot-empowering-defenders-at-the-speed-of-ai/>
- [29] “Introducing PaLM 2,” May 2023. [Online]. Available: <https://blog.google/technology/ai/google-palm-2-ai-large-language-model/>
- [30] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, and others, “Judging LLM-as-a-judge with MT-Bench and Chatbot Arena,” *arXiv preprint arXiv:2306.05685*, 2023.
- [31] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, “Gpteval: Nlg evaluation using gpt-4 with better human alignment,” *arXiv preprint arXiv:2303.16634*, 2023.
- [32] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021.
- [33] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, and others, “Training a helpful and harmless assistant with reinforcement learning from human feedback,” *arXiv preprint arXiv:2204.05862*, 2022.
- [34] M. Zhong, Y. Liu, D. Yin, Y. Mao, Y. Jiao, P. Liu, C. Zhu, H. Ji, and J. Han, “Towards a unified multi-dimensional evaluator for text generation,” *arXiv preprint arXiv:2210.07197*, 2022.
- [35] R. Liu, G. Zhang, X. Feng, and S. Vosoughi, “Aligning generative language models with human values,” in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 241–252.

## APPENDIX A PROMPTS

### A. Instructions Generation Prompt

You are asked to come up with a set of 8 diverse cybersecurity task instructions. These task instructions will be given to a LLM model and we will evaluate the LLM model for completing the instructions. Here are the requirements:

1. Try not to repeat the verb for each instruction to maximize diversity.

2. The language used for the instruction should be technical and specialized for the cybersecurity field. For example, you should combine questions with imperative instructions using cybersecurity terms.

3. The type of instructions should cover diverse types of cyber security tasks like TTP mapping in ATT&CK, vulnerability detection, risk assessment, incident response, etc.

4. A LLM language model with cyber security knowledge should be able to complete the instruction. For example, do not ask the assistant to physically access any systems or networks. For another example, do not ask the assistant to install any new hardware or software.

5. The instructions should be in English.

6. The instructions should be 1 to 2 sentences long. Either an imperative sentence or a question is permitted.

7. You should generate an appropriate input with technical details for the instruction. The input should contain system configurations, network architecture diagrams, log files, etc. to make the instruction challenging. The input should ideally not exceed 100 words.

8. Not all instructions require input. For example, when an instruction asks about some general cybersecurity concepts or frameworks, it is not necessary to provide a specific context. In this case, we simply put "<noinput>" in the input field.

9. The output should demonstrate the language model's ability to complete the cybersecurity task. Make sure the output is less than 300 words.

### B. Conversations Generation Prompt

I'll send you the text and you should try to convert it into multiple chat-style pair and return it in JSON. Remember, the first sentence must not have any ambiguous references, such as "Do you know about this vulnerability?" Instead, you should mention the specific vulnerability ID or name. The number of dialogue turns and the logical association between turns depends on the text content, taking into account contextual associations, references, and chronological order. For example:

[ "user": "What are homoglyphs?", "bot": "Homoglyphs are glyphs, pictures, or icons that have different meanings or character codes for a program than what a human user can see. They often look the same or are very similar to each other, which can lead to confusion for users.", "user": "How can adversaries exploit homoglyphs?", "bot": "Adversaries can use homoglyphs to perform phishing attacks or to create back door usernames that are visually similar to legitimate ones, making it more difficult for system administrators to detect malicious activity. They can also be used to forge log entries and steal user credentials.", "user": "How can I protect myself from homoglyph attacks?", "bot": "You can check to see if links or usernames contain homoglyphs before submitting or clicking on them. You can also use a browser that displays Punycode for IDNs in URLs, or an email client that has strict filters and prevents messages that mix character sets.", "user": "Can you give me an example of a homoglyph attack?", "bot": "An adversary can create a phishing site using a URL that contains homoglyphs, making it appear to be a trusted domain. They can also create back door usernames that are visually similar to legitimate ones, making it difficult for system administrators to detect malicious activity. For example, an adversary could create a username for a website that looks like 'admin', but replaces some of the ASCII characters with Cyrillic characters that look very similar. Any actions the adversary performs with this username will be saved to the log file and look like they came from a legitimate administrator account." ]

### C. LLMs Judgment Prompt

I will give you the following dictionary:

{ "instruction": "instruction content", "input": "input content", "model-A-output": "output content of model A", "model-B-output": "output content of model B" }

where "input" can be empty, and "model-A-output" and "model-B-output" are the responses of model A and model B to "instruction" and "input".

You need to compare "model-A-output" and "model-B-output" based on the dictionary content and choose from the following three options: A is better, B is better, or tie.

Only answer one of the given three options, without any further explanation.

### D. Profession Scenario Conversation Prompt

Design a dialogue scene that meets the following requirements:

1. There are two characters, A and B. A is a security practitioner and B is a cybersecurity-related

profession. A consults B on practical business issues that rely on B's professional knowledge;

2. The conversation content is related to B's professional skills and knowledge;

3. The dialogue content should be as rich as possible;

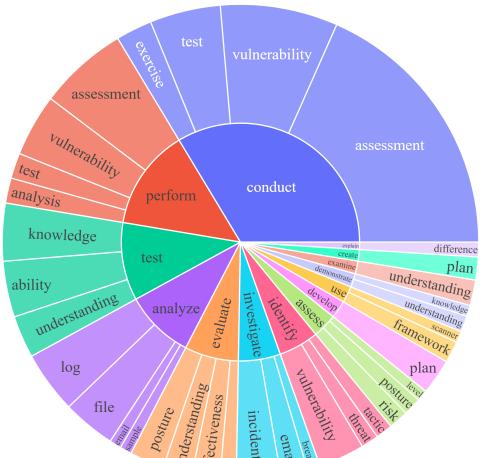
4. The more rounds of dialogue, the better, to ensure that the problem is fully discussed;

5. The discussion can be extended and expanded;

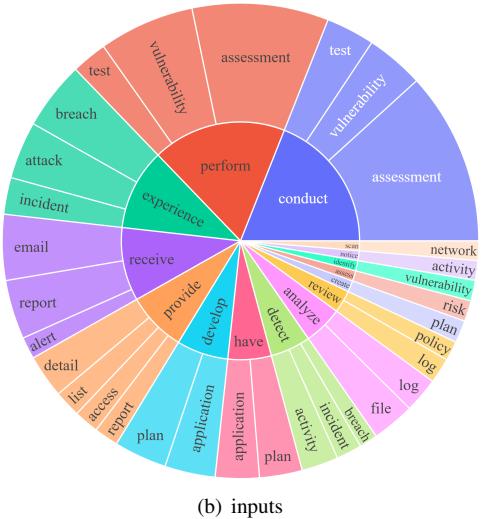
6. No current events or news will be involved.

## VI. APPENDIX B VISUALIZATION OF LLMs GENERATED INSTRUCTIONS

Fig. 8 demonstrates the diversity of our domain instructions. The inner circle of the plot represents the root verb, and the outer circle represents the direct objects.



(a) instructions



(b) inputs

Fig. 8. The top 20 most common root verbs and their top 4 direct noun objects in (a) the instructions, and (b) the inputs.

## VII. APPENDIX C CLAUDE EVALUATION COMPARISON PRINCIPLES

**User:**

I will give you the following dictionary:

{ "instruction": "instruction content", "input": "input content", "model-A-output": "output content of model A", "model-B-output": "output content of model B" }.

where "input" can be empty, and "model-A-output" and "model-B-output" are the responses of model A and model B to "instruction" and "input".

You need to compare "model-A-output" and "model-B-output" based on the dictionary content and choose from the following three options: A is better, B is better, or tie.

Do you understand?

**Claude:**

Okay, I understand. Based on the dictionary you provide, I will compare the output of Model A and Model B and choose one of the following options:

1. A is better
2. B is better
3. Tie

My approach will be:

1. Check if "model-A-output" and "model-B-output" are exactly the same. If so, return "Tie".

2. If not, I will consider the following factors to determine which model responds better:

- Content relevance: Which model response is more relevant to the "instruction" and "input" context. The more relevant and on-topic response will be considered better.
- Content coherence and logic: Which model response is more coherent, logical and flows better. The more coherent and logically consistent response will be considered better.
- Content accuracy and correctness: Which model response seems more factually accurate and correct. The more accurate response will be considered better.
- Overall content quality: A subjective evaluation of which model response is of higher quality overall based on the above factors. The higher quality response will be considered better.

3. If after evaluating the above factors I still cannot clearly determine which model is better, the result will be "Tie".

## VIII. APPENDIX D

We conducted experiments on 12 models using various parameter combinations, and the results for parted parameter combinations are shown in Table III. It can be seen that the model achieves the best performance with the combination of temperature 0.3 and context window 4.

TABLE III  
TEMPERATURE AND CONTEXT WINDOW SELECTION FOR CYBERSECURITY CONVERSATION EVALUATION

models	parameters		metrics		
	temperature	context_length	avg_chrf	avg_length_ratio	avg_bert_score
<b>text-davinci-003</b>	0.3	2	0.37	1.54	0.62
	0.3	3	0.37	1.42	0.62
	<b>0.3</b>	<b>4</b>	<b>0.37</b>	<b>1.42</b>	<b>0.63</b>
	0.9	2	0.35	1.66	0.60
	0.9	3	0.35	1.63	0.61
	0.9	4	0.36	1.59	0.61
<b>gpt-3.5-turbo</b>	<b>0.3</b>	<b>1</b>	<b>0.43</b>	<b>2.62</b>	<b>0.64</b>
	0.3	4	0.42	2.72	0.64
	0.6	2	0.42	2.82	0.64
	0.6	4	0.42	2.77	0.64
	0.9	1	0.43	2.62	0.64
	0.9	2	0.40	2.79	0.63
	0.9	3	0.40	2.80	0.63
<b>Llama-7b</b>	0.3	2	0.21	0.83	0.56
	<b>0.3</b>	<b>4</b>	<b>0.24</b>	<b>0.87</b>	<b>0.58</b>
	0.6	1	0.17	0.66	0.50
	0.6	2	0.22	0.74	0.54
	0.6	4	0.24	0.87	0.56
	0.9	2	0.23	1.07	0.52
	0.9	4	0.25	1.10	0.54
<b>Llama-7b-lora-iio</b>	0.3	1	0.25	0.8	0.57
	0.3	2	0.28	0.84	0.60
	<b>0.3</b>	<b>4</b>	<b>0.29</b>	<b>0.83</b>	<b>0.61</b>
	0.6	1	0.25	0.87	0.56
	0.6	2	0.27	0.91	0.58
	0.9	1	0.25	1.07	0.54
	0.9	2	0.28	1.07	0.57
<b>Llama-7b-lora-turn</b>	0.3	1	0.27	0.84	0.59
	0.3	3	0.31	0.88	0.62
	<b>0.3</b>	<b>4</b>	<b>0.32</b>	<b>1.08</b>	<b>0.62</b>
	0.6	2	0.30	0.64	0.59
	0.6	3	0.31	0.90	0.61
	0.6	4	0.31	0.92	0.60
	0.9	2	0.28	1.00	0.57
<b>Llama-7b-lora-turn</b>	0.9	3	0.29	1.09	0.58
	0.9	4	0.31	1.07	0.60