Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.073979

ARTICLE

TinySecGPT: Small-Parameter LLMS Can Outperform Large-Parameter LLMS in Cybersecurity

Anfeng Yang, Fei Kang and Wenjuan Bu*

Information Engineering University, Zhengzhou, 450001, China

*Corresponding Author: Wenjuan Bu. Email: buwenjuan_521@163.com

Received: 29 September 2025; Accepted: 02 December 2025; Published: 12 March 2026

ABSTRACT: Large language models (LLMs) have demonstrated significant capabilities in semantic understanding and code generation. However, cybersecurity tasks often require prompting the adaptation of open-source models to this domain. Despite their effectiveness, large-parameter LLMs incur substantial memory usage and runtime costs during task inference and downstream fine-tuning for cybersecurity applications. In this study, we fine-tuned six LLMs with parameters under 4 billion using LoRA (Low-Rank Adaptation) on specific cybersecurity instruction datasets, employing evaluation metrics similar to Hackmentor. Results indicate that post-fine-tuning, smaller models achieved victory or parity rates up to 85% against larger models like Qwen-1.5-14B on cybersecurity test datasets, with the best model reaching a 90% win or tie rate compared to SecGPT. Additionally, these smaller models required significantly less computational resources, reducing fine-tuning times by up to 53% and enhancing efficiency in downstream tasks. Further validation showed that with minimal fine-tuning, our models achieved a performance gain of 21.66% to 31.32% in tactical extraction and 30.69% to 40.42% in technical extraction tasks, significantly outperforming ChatGPT. These findings highlight the potential of smaller parameter LLMs for optimizing performance and resource utilization in cybersecurity applications including methods such as technique and tactic extraction. It will facilitate future research on the application of small-parameter large language models in the cybersecurity domain.

KEYWORDS: Tinyllm; fine-tuning time; Elorating; SecGPT; TIME COST; cybersecurity downstream tasks

1 Introduction

Artificial intelligence (AI) technologies, such as machine learning, deep learning, and natural language processing (NLP), have demonstrated considerable potential in enhancing capabilities across threat detection, behavioral analysis, and automated response systems [1]. ChatGPT showed the remarkable ability of large language models to follow human instructions [2]. However, although the vertical domain large language model can show strong general problem processing ability on related domain tasks, due to the large number of parameters, it will produce huge memory overhead in the actual reasoning and fine-tuning process, and it is difficult to re-fine-tune the instruction on downstream specific tasks. The large language model represented by Phi-2 [3] proves that a model with a not large enough number of parameters can also show strong task processing capability on general tasks, and its effect is comparable to that of a large language model with several times more parameters. The subsequent Phi-3 [4] model is even more comparable to the related large parameter model with a small number of parameters. Due to the network security domain models constructed in previous studies are all for large language models with more than 7b parameters. Incremental pre-training or command fine-tuning can make them capable of answering network security domain knowledge to a certain extent. However, due to the large number of parameters, it will occupy a huge memory when reasoning on them, and it is difficult to perform command fine-tuning on downstream network security tasks. This creates a huge memory overhead. Therefore, the objective of this paper is to fine-tune lora in the field of network security for a large language model with less than 400 million parameters, test it with the relevant data set in the paper [5], and compare the effect of 145 relevant test indicators with that of the large model SecGPT customized in a specific field of network security. The experimental results show that, even with a much smaller number of parameters than these network security domain models, our model can still achieve similar results on the relevant network security domain metrics, while significantly reducing the runtime overhead. The structure of our paper is as follows: Section 1 introduces the background and context of this study; Section 2 reviews related work, including the origins of large language models and their applications; Section 3 elaborates on the research questions addressed by our method; Section 4 details the experimental setup and process; Section 5 presents the study's conclusion and discusses the relevant findings. This paper presents the conceptual framework for comparing lightly-tuned compact LLMs in cybersecurity, with the schematic diagram shown in Fig. 1.

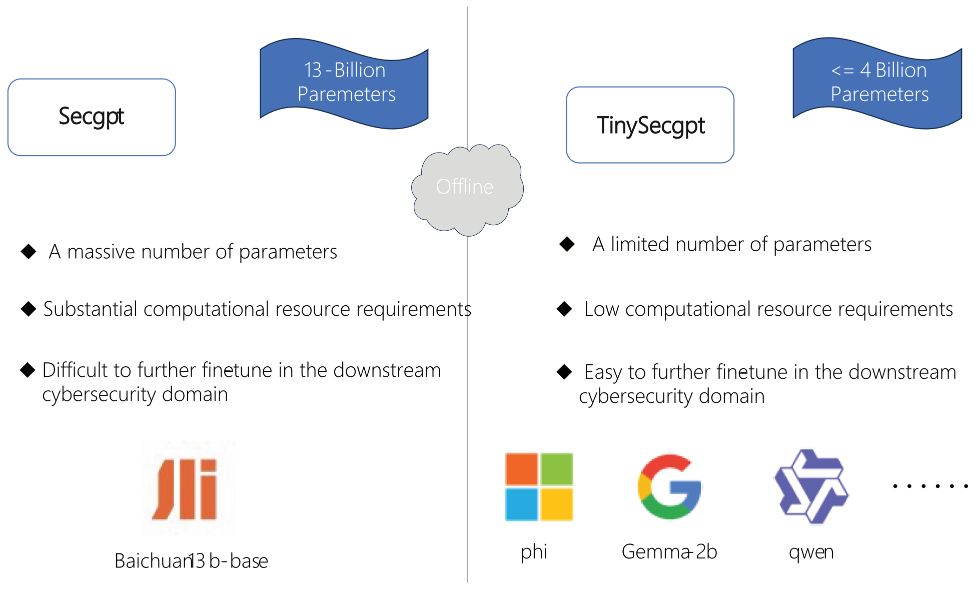

Secgpt
13-Billion Paremeters
TinySecgpt
<= 4 Billion Paremeters
Offline
◆ A massive number of parameters
◆ Substantial computational resource requirements
◆ Difficult to further finetune in the downstream cybersecurity domain
◆ A limited number of parameters
◆ Low computational resource requirements
◆ Easy to further finetune in the downstream cybersecurity domain
Baichuan13b-base
phi
Gemma-2b
qwen
......

Figure 1: Compare scheme

The contributions of this article are mainly as follows:

- To the best of our knowledge, this work represents the first attempt to fine-tune small-parameter large language models for cybersecurity applications.
- This paper systematically compares the performance of six sub-4B parameter large language models after instruction fine-tuning on cybersecurity-related tasks.
- After fine-tuning, our compact large language model achieves performance comparable to 14B-parameter models on standard cybersecurity benchmarks, even approaching the capability of specialized security LLMs like SecGPT.
- Among our constructed models, the smallest large language model contains only one-third the parameters of the largest existing cybersecurity-specialized LLM. Under the same framework, it demonstrates significantly faster fine-tuning speeds compared to larger models while better adapting to downstream cybersecurity task requirements.

- Large language models with a small number of parameters can outperform ChatGPT by 21.66% to 31.32% on tactical extraction tasks and by 30.69% to 40.42% on technical extraction tasks after only simple fine-tuning, demonstrating significant advantages in comparison.

2 Related Works

2.1 Large Language Models in Cybersecurity

Large language models show strong ability on relevant general tasks, among which, the quality and quantity of pre-trained corpus, the number of parameters of the model and the number of parameters activated during inference can represent the processing ability of large language models themselves on general tasks to a certain extent. For the field of network security, the related tasks are highly private, and the use of closed-source large language models like GPT-4 to complete the tasks in the field of network security not only requires complex prompt word engineering guidance, but also faces the problem of sensitive privacy data leakage. Therefore, some researches aimed at this problem, fine-tune or re-pre-train on the open source large language model, so that it can be applied to the related network security field tasks, and achieve better results. Hackmentor was one of the first to start this research. It first set up seed fine-tuning data for tasks in the field of network security and used the method of self-instruct [6]. Based on this, a large number of relevant instruction fine-tuning data sets were generated. Then the lora instruction is fine-tuned by several large language models with different parameters such as llama [7]. Subsequently, on the basis of Baichuan-13b-base, SecGPT re-collected extensive data such as books and vulnerability analysis reports in the field of network security and re-pre-trained them. The trained large language model performs well in the field of network security tasks.

2.2 Other AI Approaches in Cybersecurity

Beyond the extensive applications of Large Language Models (LLMs) in cybersecurity, numerous traditional AI methods have also been widely employed. For instance, prior research has utilized hybrid machine learning models to automatically identify relevant threats on hacker forums [8], developed systems for discovering cyber threat intelligence on Twitter [9], applied deep neural networks to process security-related information from social media [10], and introduced methods for extracting low-level threat actions from open-source CTI [11]. Some studies employ an automated approach to extract low-level adversarial behaviors from public CTI sources, which is foundational for enabling timely defensive decision-making [12]. There is also a body of research [13–16] focusing on content analysis within the field of threat intelligence.

2.3 Large Language Models with Small Parameter Number

Increasing the quantity and quality of pre-training data enhances the performance of smaller-scale [17] large language models. Knowledge distillation from large language models to small language models demonstrates the significant potential of the latter [18]. Some studies have designed pre-trained large language models with fewer parameters specifically for Chinese [19].

Compared with the related models with a large number of parameters, the large language model with a small number of parameters has less overhead for inference and fine tuning, and is easier to adapt to the specific tasks in the field of network security.

3 Methodology

This chapter details the origin of the instruction fine-tuning dataset employed in our approach and outlines the associated testing methodology with the overall framework illustrated in Fig. 2.

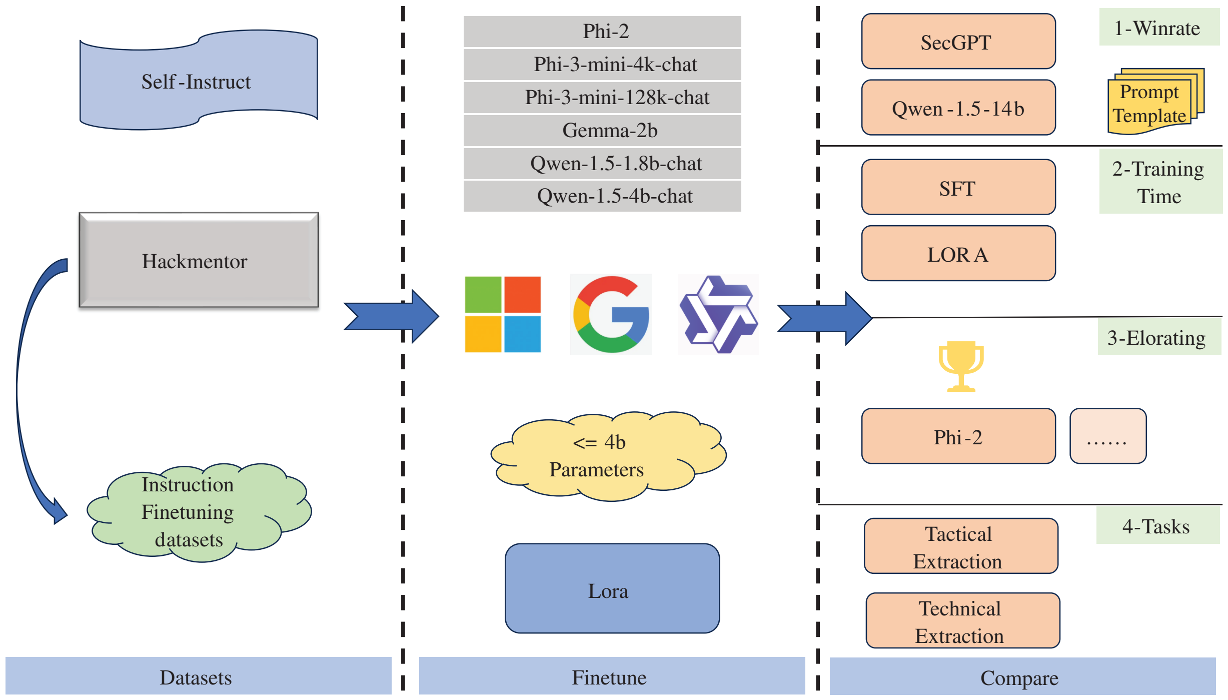


Figure 2: Total scheme

3.1 Dataset Collection and Experimental Methodology

This study adopts the training and testing dataset from paper [5], which was constructed using the self-instruct methodology. Specifically, the dataset was generated by fine-tuning GPT-3.5-turbo on 145 seed instructions, yielding an augmented dataset of over 10,000 instruction-following samples. Furthermore, it is noteworthy that the resulting training set of over 10,000 examples explicitly excluded the original 145 seed instructions. The corpus of over 10,000 instructions was generated based on 145 seed instructions, with the generated set excluding these original seed instructions. The generated dataset exhibits a ROUGE score below 0.6 when compared to the original seed instructions, ensuring sufficient diversity while maintaining relevance. Consequently, this dataset serves as the benchmark evaluation criterion in our experiments.

To assess the performance of different models, we employ DeepSeek-V2 as the evaluator, which has demonstrated superior performance over GPT-4 in relevant benchmarks. The evaluation is conducted using the prompt template provided in Appendix A, with outcomes classified as either a win for one model or a tie.

3.2 Comparative Analysis Approach

Based on the requirements of the model with more than 4B parameters in this paper, phi-2, Phi3-mini-4K-Chat, Phi3-mini-128K-Chat, and gemma-2b were selected, respectively. Six different large language models with small parameter numbers, namely qwen-1.5-1.8b-chat and qwen-1.5-4b-chat, were fine-tuned with lora on more than 10,000 datasets to conduct in-depth mining of the capabilities of the related models. Meanwhile, it is also aimed to verify how the fine-tuned small parameter model performs on the fine-tuned dataset in the field of network security. For the final fine-tuned model, not only is its performance compared with the general large language model with 14b parameters that has not been fine-tuned, but also with the large language model that has been re-pre-trained and fine-tuned specific to the field of network security. Finally, a horizontal comparison is made among the models with different types of small parameters that have been fine-tuned with data in the field of network security. And the advantages and disadvantages of the effects among different models are judged through the EloRating index.

4 Experimental Framework and Systematic Evaluation

4.1 Compared with General Large Language Models

To determine the effect of the small parameter model after re-fine-tuning in the field of network security.

4.1.1 Comparison with the Performance of Large Language Models in General Domains

To determine the effect of the small parameter model after re-fine-tuning in the field of Cyber security. The detailed result is presented in Table 1. The values listed beneath each model in the table represent their performance on the set of 145 planting instructions.

Table 1: Compare with the performance of large language models in general domains

Model name	Custom-model	tie	qwen1.5-14B
qwen-1.5-4b-chat	10	98	37
qwen-1.5-1.8b-chat	6	95	44
phi-3-mini-4k-chat	22	97	26
phi-3-mini-128k-chat	40	39	66
Phi-2	8	81	56
gemma-2b	29	95	21

4.1.2 Compared with Large Models in the Field of Cyber Security

This chapter presents the performance evaluation of TinySecGPT on cybersecurity-related test datasets, comparing it against two types of models: (1) large-scale general-purpose language models with 14B parameters (e.g., Qwen-1.5-14B), and (2) cybersecurity-specific large language models that have been retrained and fine-tuned on domain-specific data. The experimental results demonstrate that properly fine-tuned small-parameter models can achieve competitive performance—matching or surpassing the general-purpose Qwen-1.5-14B model in up to 85% of test cases. Notably, the best-performing variant even achieves comparable or superior results to SecGPT (a specialized cybersecurity LLM) in 90% of evaluations (see Table 2). Our experiments demonstrate that sub-4B LLMs, when specifically trained for cybersecurity, can achieve performance comparable to not only general-purpose 14B LLMs but also other large models that are specially trained for cybersecurity in certain scenarios.

Table 2: Compare with the performance of large language models in cyber security

Model name	Custom-model	tie	SecGPT
qwen-1.5-4b-chat	25	81	39
qwen-1.5-1.8b-chat	31	63	51
phi-3-mini-4k-chat	56	75	14
phi-3-mini-128k-chat	50	68	27
Phi-2	64	26	55
gemma-2b	23	58	64

4.2 The Time Cost on the Same Dataset

4.2.1 Lora Training Overhead

To visually verify the comparison of the time spent on full-parameter fine-tuning between a language model with a large number of small parameters and a language model with a large number of large parameters, this paper adopts the same fine-tuning framework to compare the time cost of relevant language models with a large number of small parameters and the large-parameter model that undergoes full-parameter fine-tuning in the paper [5]. Table 3 summarizes our main results. Parameters in this table are measured in billions, while the durations for SFT and LoRA are reported in seconds.

Table 3: Lora training overhead

Model name	Parameters (b)	Lora fine-tuning (s)
phi-2	2.7	1408
qwen1.5-14b	14	3422
qwen-1.5-1.8b-chat	1.8	867
qwen-1.5-4b-chat	4	1508
phi-3-mini-128k-chat	2.7	1797
phi-3-mini-4k-chat	2.7	1623
gemma-2b	2	913
Llama-7b	7	2239
Llama-13b	13	3903
Vicuna-13b-1.1	13	4549
Vicuna-7b-1.1	7	2630

This paper conducts tests on two A100s. The relevant hyperparameter ratios are as follows: per_device_train_batch_size: 3, gradient_accumulation_steps: 2, learning_rate: 0.0001, cutoff_len = 1024.

4.2.2 Comparison of SFT Fine-Tuning Time Cost

To visually verify the comparison of the time spent on full-parameter fine-tuning between a language model with a large number of small parameters and a language model with a large number of large parameters, this paper adopts the same fine-tuning framework to compare the time cost of relevant language models with a large number of small parameters and the large-parameter model that undergoes full-parameter fine-tuning in the paper [5]. This paper conducts tests on two A100s. Table 4 summarizes our main results. The relevant hyperparameter ratios are as follows: per_device_train_batch_size: 3, gradient_accumulation_steps: 2, learning_rate: 0.0001.

Table 4: Full-parameter fine-tuning

Model name	Parameters (b)	SFT (s)
phi-2	2.7	6100
qwen-1.5-1.8b-chat	1.8	4725
qwen-1.5-4b-chat	4	8241
phi-3-mini-128k-chat	2.7	5773
phi-3-mini-4k-chat	2.7	5668
gemma-2b	2	4015
Llama-7b	7	11,030
Llama-7b	7	10,608

4.3 The Effect Comparison among Large Language Models with Small Number of Parameters

Models with a small number of parameters have certain performance capabilities in related test problems. Meanwhile, in order to systematically compare the effects of large language models with a small number of parameters on specific domain tasks, this paper compares the effects of several general large language models and evaluates them using an Elo rating metric similar to that in [5] to determine their scores on relevant metrics. The Elo rating system quantifies the relative skill of agents by analyzing competitive outcomes, dynamically updating scores based on performance against expectations. The core update formula is $R_{\text{new}} = R_{\text{old}} + K \times (S - E)$, where R_{new} and R_{old} are the updated and previous ratings, K is a constant controlling the adjustment magnitude, S is the actual game outcome (1 for win, 0.5 for draw, 0 for loss), and E is the expected score calculated from the players' pre-game ratings.

Comparison of the effects of large language models related to effects under relevant reasoning frameworks:

To further provide relevant researchers with the performance of large language models with a small number of parameters in the relevant evaluation indicators in the field of network security, this paper adopts the Elo rating method to evaluate the performance comparison among related large language models with a small number of parameters and determine their Elo rating, thereby providing inspiration for subsequent researchers. A prominent method for evaluating the performance of Large Language Models (LLMs), particularly in open-ended and conversational tasks, is the Win Rate, defined as the proportion of victories in pairwise comparisons: Win Rate = $\frac{N_{\text{win}}}{N_{\text{total}}}$, where N_{win} is the number of wins and N_{total} is the total number of pairwise comparisons. This metric serves as a relative and pairwise performance measure that directly quantifies human or model preference between two competing systems. A comparative analysis of the win rates for the evaluated models that have been fine-tuned is summarized in Fig. 3.

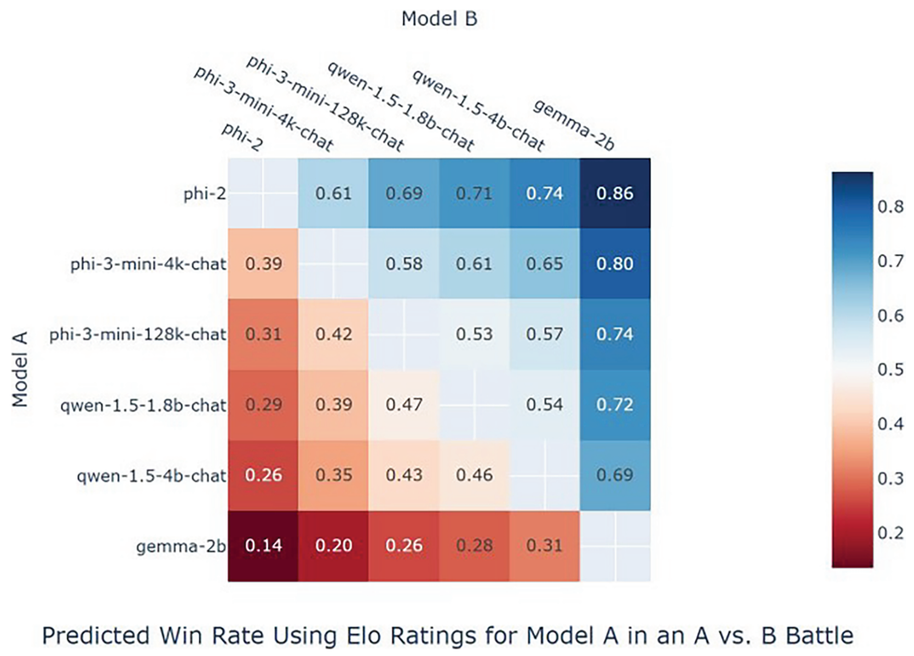


Figure 3: Win-rate comparison

The relevant results of the Elorating judgment are shown in Table 5. It can be seen that a certain model performs the best in the EloRating indicator. There should be two possibilities for judging it. This model originally performed well in the relevant test indicators. Through this fine-tuning, it will perform better in the related downstream problems. Therefore, this also proves that when completing related downstream tasks, it is possible to consider using large language models with relevant small parameter quantities. For example, the phi-3 series models can complete many tasks such as tactical and technical extraction in the field of network security and named entity recognition in the field of network security on the fine-tuned models.

Table 5: Elo-rating

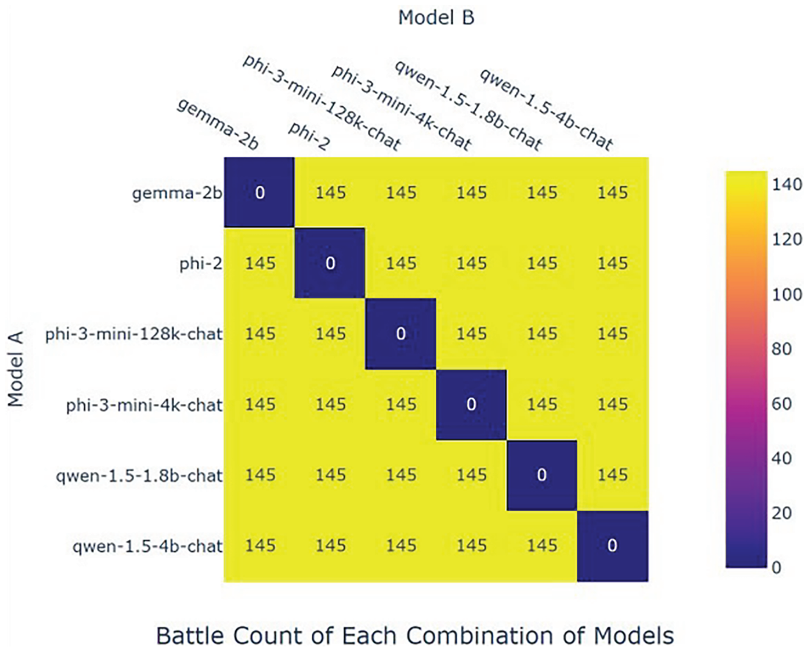
Name	Model	Elo-rating
0	phi-2	1146
1	phi-3-mini-4k-chat	1068
2	phi-3-mini-128k-chat	1009
3	qwen-1.5-1.8b-chat	989
4	qwen-1.5-4b-chat	961
5	gemma-2b	825

Ultimately, a comparative verification was conducted on the task benchmarks in the relevant field of network security with large language models such as SecGPT, Hackmentor, and SecGPT-1.5, which were re-pre-trained, incrementally trained, or fine-tuned for relevant instructions in the field of network security.

The final experimental results can also indicate that The fine-tuned model can not only show better results in the evaluation of related tasks in the field of network security on the basis that the number of parameters is much smaller than that of other large language models in the field of network security. Not only that, but it can also be fine-tuned and re-pre-trained more conveniently on downstream tasks. Therefore, the work of this paper has certain reference and comparative significance for the subsequent related specific tasks applied in the field of network security.

Furthermore, in this experiment, for the first time, the performance of a large language model with a small number of parameters that has been fine-tuned through the instruction fine-tuning dataset in the field of network security is compared with that of related models. This enables a systematic comparison of the performance and related capabilities of large language models with a small number of parameters in answering questions in the field of network security. Thus, it can be seen that large language models with different types of small parameter numbers also have different statements when answering questions related to the field of network security. The reason for this might be that the pre-trained corpora are different, which contain different distributions of pre-trained corpora. For large language models with relatively good performance, It might be that a considerable amount of content related to the field of cyber security was added to the pre-trained corpus.

The experiment adopted the methods related to hackementor and used its evaluation prompts. Advanced large language models such as the most advanced DeepSeek-V2 were called respectively for evaluation. Eventually, on the relevant 145 questions, after lora fine-tuning, the relevant models showed size comparisons similar to the number of their parameters. However, it can demonstrate relatively good results on related issues. The number of test seed instructions is shown in Fig. 4.



Model A \ Model B	gemma-2b	phi-2	phi-3-mini-128k-chat	phi-3-mini-4k-chat	qwen-1.5-1.8b-chat	qwen-1.5-4b-chat
gemma-2b	0	145	145	145	145	145
phi-2	145	0	145	145	145	145
phi-3-mini-128k-chat	145	145	0	145	145	145
phi-3-mini-4k-chat	145	145	145	0	145	145
qwen-1.5-1.8b-chat	145	145	145	145	0	145
qwen-1.5-4b-chat	145	145	145	145	145	0

Figure 4: Battle counts comparison

The cases with divergent outcomes are presented in Fig. 5, while the comparable results among the six fine-tuned smaller LLMs are shown in Fig. 6.

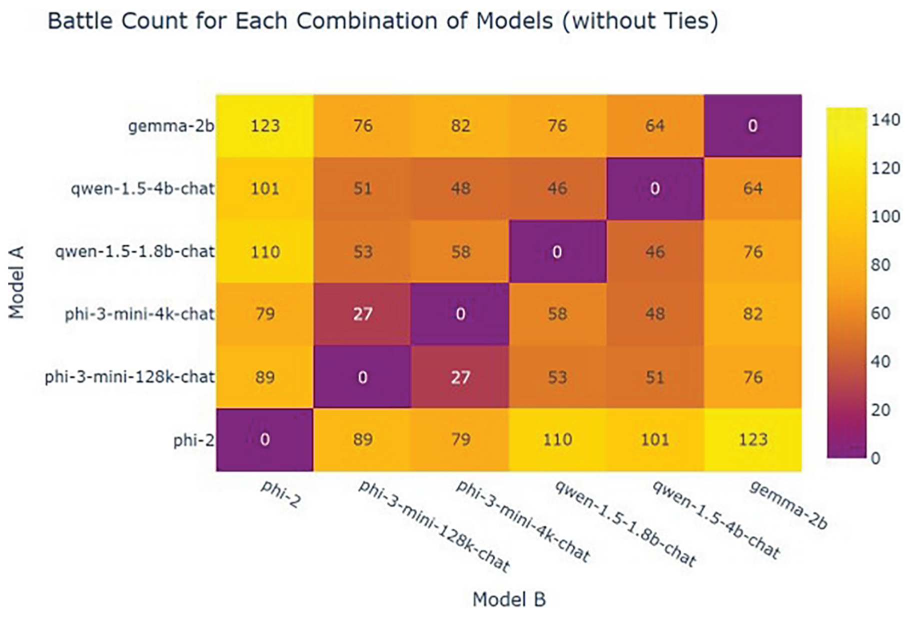


Figure 5: Battle counts without tie

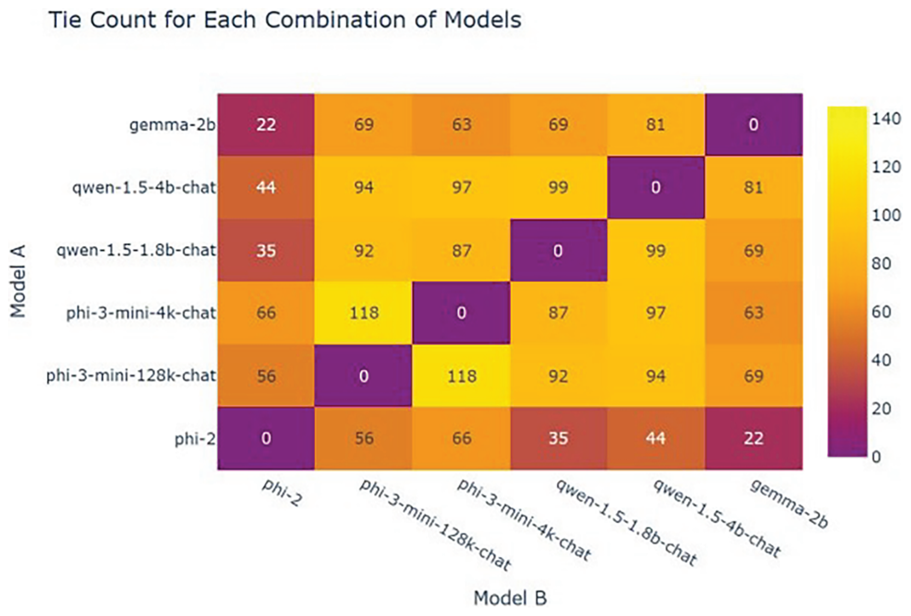


Figure 6: Battle counts

First, directly compare the output effect of the model with llama3-70b, so as to compare its performance in answering related questions in the field of network security.

Subsequently, the effects of the pairwise small parameter number models on the generation of network security will be compared. For six different models with small parameter quantities, pairwise comparisons can be made, and thus the relevant confusion matrix diagrams can be obtained in total. From this, we can also see that different models give different answers to related questions. This might be due to the different pre-trained corpora, which leads to their different focuses on capabilities in related tasks in the field of network security. Finally, the Elo ratings of different fine-tuned large language models with a small number of parameters were compared, and the relevant contents were sorted according to their scores from high to low, as follows:

Regarding the specific steps of the experiment, first, its response effect on the original data was compared with that of Hackmentor, and SecGPT was also compared with its effect on the relevant datasets of hackmentor. By comparison, it can be seen that the fine-tuned relevant small language models can achieve corresponding effects on related complex tasks and have corresponding advantages compared with hackmentor. Experimental comparisons were conducted respectively on the instruction fine-tuning datasets with a large ROUGHE distance from the instruction fine-tuning dataset, namely the 145-instruction fine-tuning dataset, which can demonstrate its certain advantages. It was compared respectively with Hackmentor, SecGPT, and SecGPT-1.5. Finally, the specific values of the Win rate related to various models of the relevant predictions are presented in the Fig. 3.

Secondly, by comparing the contents among them, it can be seen that phi3-mini performs the best in the relevant effects and should be considered to have a relatively strong pre-training ability. But at the same time, its parameter number is much smaller than that of the relevant network security models, and the parameter numbers of the other models are all related to 7b or 14b.

For the performance of the relevant large language model, the content related to balancing in the model is specifically shown in Figs. 5 and 6.

4.4 Downstream Task Generalization Analysis

The previous text discussed the effect on general network security tasks. Not only that, we also found that large language models with small parameters also demonstrated good performance in related downstream tasks.

This paper tests the effect of the relevant small parameter model on the relevant downstream tactical technology extraction datasets. After fine-tuning with lora, the relevant effects are tested respectively on the relevant tactical extraction datasets and the technical extraction datasets.

This method tests the performance of several large language models with small parameters in the experiment on the same training set, validation set and test set, respectively, and directly compares the performance with that of the more capable closed-source model chatgpt. The relevant accuracy was tested respectively from the perspectives of tactical extraction, technical extraction, etc. Experiments have proved that these models with parameters smaller than 4b can demonstrate outstanding advantages in related tactical and technical data extraction after simple fine-tuning, and can be better applied to related tasks in the downstream network security field.

The complete tactical extraction dataset is derived from reference [20], which contains a total of 26,602 datasets and follows the rule of 8:1: The proportion of 1 is divided into the training set, the validation set, and the test set. This method converts them into the corresponding fine-tuning data set of large model instructions, using the alpaca format, for phi-3-mini-4k-chat, phi-3-mini-128k-chat, and qwen1.5-1.8b-chat, respectively. qwen1.5-4b-chat, Phi-2, Gema-2B (unsloth), six large language models with parameters less than 4b were fine-tuned with lora, and the experimental setup was tested on 50%A100. It took about 3 to 4 h. Compared with other large language models with a large number of parameters, the time spent on fine-tuning and the related video memory overhead were much less than those of the relevant models. Specifically, its related parameters are as follows:

4.4.1 Tactical Extraction Effect

To verify the effect of tactical extraction, in this paper, lora fine-tuning was performed on three large language models with relevant parameters less than 4b on the relevant training set, namely phi-3-mini-4k-chat, phi-3-mini-128k-chat, and qwen1.5-1.8b-chat. For all LoRA fine-tuning experiments, the training batch size was set to 2, the learning rate was 1.0e−4, and the models were trained for 3.0 epochs. All experiments were performed on an A100 GPU with a memory footprint limited to 50%.

The model was waited for and tested on the final result. First is the tactical extraction effect, and its specific effect is as follows: It can be seen from the relevant experiments that large language models with small parameters outperform Chatgpt by 21.66% to 34.48%, respectively, and have a relatively obvious advantage in the effect of tactical extraction (see Table 6).

Table 6: TACTIC precision

Model-name	Parameters (billion)	Precision
phi-3-mini-4k-chat	3.8	73.92%
phi-3-mini-128k-chat	3.8	64.26%
qwen1.5-1.8b-chat	1.8	67.34%
qwen1.5-4b-chat	4	77.08%
Phi-2	2.7	66.10%
Gemma-2b (unsloth)	2	68.13%
Chatgpt	Almost 175	42.60%

4.4.2 Technical Extraction Effect

Meanwhile, to verify the extraction effect of the relevant technology, this experiment also fine-tuned lora on three large language models with parameters less than 4b and then conducted tests on downstream tasks [16].

For the relevant indicators of technical extraction, the specific effect is shown in the Table 7. It can be seen that the small-parameter model, with only 46 to 97 times the number of parameters of Chatgpt, only needs a small amount of fine-tuning. The performance of the small-parameter large-language model is approximately 30.69% to 42.85% higher than that of Chatgpt. It has relatively obvious effect advantages (see Table 7).

Table 7: Techinque precision

Model-name	Parameters (billion)	Precision
phi-3-mini-4k-chat	3.8	58.32%
phi-3-mini-128k-chat	3.8	48.82%
qwen1.5-1.8b-chat	1.8	48.59%
qwen1.5-4b-chat	4	59.75%
Phi-2	2.7	46.86%

(Continued)

Table 7 (continued)

Model-name	Parameters (billion)	Precision
Gemma-2b (unsloth)	2	49.06%
Chatgpt	Almost 175	17.90%

Secondly, by comparing the contents among them, it can be seen that phi3-mini performs the best in the relevant effects and should be considered to have a relatively strong pre-training ability. But at the same time, its parameter number is much smaller than that of the relevant network security models, and the parameter numbers of the other models are all related to 7b or 14b.

5 Discussion and Conclusion

5.1 Discussion

LLM capabilities are a function of architecture, training data, and parameters—not parameters alone. The strong performance of models like phi-2, trained on extensive corpora, consequently explains why certain compact models are competitive in NLP-based cybersecurity tasks. Future research should focus on analyzing more in-depth NLP tasks in cybersecurity, including Named Entity Recognition, to further explore the application potential of these compact models in the cybersecurity domain.

5.2 Conclusion

The experiments conducted in this study demonstrate that smaller-scale large language models (LLMs), when fine-tuned on cybersecurity domain-specific downstream task datasets, can achieve comparable or even superior performance to certain proprietary LLMs. This advantage is evident not only in the comprehension of cybersecurity knowledge but also including tactics and techniques extraction in the cybersecurity domain.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Anfeng Yang bore primary responsibility for the research and manuscript preparation. Wenjuan Bu and Fei Kang both contributed significantly to the intellectual content through critical revision of the manuscript. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Appendix A

The gemma-2-b model weighted links from: https://huggingface.co//unsloth/gemma-2b.

Prompt for Deepseekv2: "I will give you the following dictionary: "instruction": "instruction content", "input": "input content", "model-A-output": "output content of model", "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”. The specific contents of the dictionary are shown below: “text”. 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.”

References

1. Ali S, Wang J, Leung VCM. AI-driven fusion with cybersecurity: exploring current trends, advanced techniques, future directions, and policy implications for evolving paradigms—a comprehensive review. Inf Fusion. 2025;118(3):102922. doi:10.1016/j.inffus.2024.102922.
2. Ouyang L, Wu J, Jiang X, Almeida D, Wainwright CL, Mishkin P, et al. Training language models to follow instructions with human feedback. In: Proceedings of the 36th International Conference on Neural Information Processing Systems; 2022 Nov 28–Dec 9; New Orleans, LA, USA. Red Hook, NY, USA: Curran Associates Inc.; 2022. p. 27730–44.
3. Li Y, Bubeck S, Eldan R, Giorno AD, Gunasekar S, Lee YT. Textbooks are all you need II: phi-1.5 technical report. arXiv:2309.05463. 2023.
4. Abdin MI, Jacobs SA, Awan AA, Aneja J, Awadallah A, Awadalla H, et al. Phi-3 technical report: a highly capable language model locally on your phone. arXiv:2404.14219. 2024.
5. Zhang J, Wen H, Deng L, Xin M, Li Z, Li L, et al. Hackmentor: fine-tuning large language models for cybersecurity. In: Proceedings of the 22nd IEEE International Conference on Trust, Security and Privacy in Computing and Communications; 2023 Nov 1–3; Exeter, UK. p. 452–61.
6. Wang Y, Kordi Y, Mishra S, Liu A, Smith NA, Khashabi D, et al. Self-instruct: aligning language models with self-generated instructions. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2023 Jul 9–14; Toronto, ON, Canada. Stroudsburg, PA, USA: Association for Computational Linguistics; 2023. p. 13484–508.
7. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al. Llama: open and efficient foundation language models. arXiv:2302.13971. 2023.
8. Deliu I, Leichter C, Franke K. Collecting cyber threat intelligence from hacker forums via a two-stage, hybrid process using support vector machines and latent dirichlet allocation. In: Proceedings of the 2018 International Conference on Big Data; 2018 Dec 10–13; Seattle, WA, USA. p. 5008–13.
9. Mittal S, Das PK, Mulwad V, Joshi A, Finin T. CyberTwitter: using Twitter to generate alerts for cybersecurity threats and vulnerabilities. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM); 2016 Aug 18–21; San Francisco, CA, USA. p. 860–7.
10. Le Sceller Q, Karbab EB, Debbabi M, Iqbal F. SONAR: automatic detection of cyber security events over the twitter stream. In: Proceedings of the 12th International Conference on Availability, Reliability and Security; 2017 Aug 29–Sep 1; Reggio Calabria, Italy. p. 1–11.
11. Dionisio N, Alves F, Ferreira PM, Bessani A. Cyberthreat detection from twitter using deep neural networks. In: Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN); 2019 Jul 14–19; Budapest, Hungary. p. 1–8.
12. Husari G, Niu X, Chu B, Al-Shaer E. Using entropy and mutual information to extract threat actions from cyber threat intelligence. In: Proceedings of the 2018 IEEE International Conference on Intelligence and Security Informatics (ISI); 2018 Nov 9–11; Miami, FL, USA. p. 1–6.
13. Zhao J, Yan Q, Liu X, Li B, Zuo G. Cyber threat intelligence modeling based on heterogeneous graph convolutional network. In: Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020); 2020 Oct 14–16; Online. San Sebastian, Spain: USENIX Association. p. 241–56.
14. Husari G, Al-Shaer E, Ahmed M, Chu B, Niu X. TTPDrill: automatic and accurate extraction of threat actions from unstructured text of CTI sources. In: Proceedings of the 33rd Annual Computer Security Applications Conference; 2017 Dec 4–8; Orlando, FL, USA. New York, NY, USA: Association for Computing Machinery; 2017. p. 103–15.

15. Niakanlahiji A, Safarnejad L, Harper R, Chu BT. IoCMiner: automatic extraction of indicators of compromise from twitter. In: Proceedings of the 2019 IEEE International Conference on Big Data (Big Data); 2019 Dec 9–12; Los Angeles, CA, USA. p. 4747–54.
16. Zhu Z, Dumitras T. ChainSmith: automatically learning the semantics of malicious campaigns by mining threat intelligence reports. In: Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P); 2018 Apr 24–26; London, UK. p. 458–72.
17. Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, Child R, et al. Scaling laws for neural language models. arXiv:2001.08361. 2020.
18. Jun J, Kim WK, Na H, Woo H, Kim J. Aspect-augmented distillation of task-oriented dialogues to small language models. Expert Syst Appl. 2025;302(1):130494. doi:10.1016/j.eswa.2025.130494.
19. Li DW, Zhong ZY, Sun YF, Shen JY, Ma ZZ, Yu CY, et al. LingLong: a high-quality small-scale chinese pre-trained language model. J Comput Res Dev. 2025;62(3):682–93. (In Chinese). doi:10.7544/issn1000-1239.202330844.
20. Li L, Huang C, Chen J. Automated discovery and mapping ATT&CK tactics and techniques for unstructured cyber threat intelligence. Comput Secur. 2024;140:103815. doi:10.1016/j.cose.2024.103815.