CyLLM-DAP: Cybersecurity Domain-Adaptive Pre-Training Framework of Large Language Models

Khang Mai, Razvan Beuran and Naoya Inoue
Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan

Keywords: Large Language Models (LLMs), Domain-Adaptive Pre-Training, Cybersecurity-Specific LLMs, MITRE Att&CK, LLM4Security.

Abstract: Recently, powerful open-source models LLMs, such as Llama 3, have become alternatives to commercial ones, especially in sensitive or regulated industries. In cybersecurity, most LLM utilization relies on custom fine-tuning or post-training methods, such as prompt engineering. Although domain-adaptive pre-training has been proven to improve the model's performance in the specialized domain, it is less used in cybersecurity due to the cumbersome implementation effort. This paper introduces CyLLM-DAP, a framework for expediting the domain specialization process of LLMs in cybersecurity by simplifying data collecting, preprocessing, and pre-training stages in low-resource settings. We demonstrate how CyLLM-DAP can be utilized to collect, process data, and develop cybersecurity-specific LLMs (CyLLMs) based on state-of-the-art open-source models (Llama 3 and Mistral v0.3). The effectiveness of domain-adaptive pre-training is confirmed via two experiments for text classification and Q&A tasks. Our evaluation results show that, when compared with general base or instruct models, injecting the LLMs with cybersecurity knowledge allows the models to generally perform better in every fine-tuning epoch for the text classification task; and brings a performance gain of up to 4.75% for the Q&A task (comparable to domain-adaptive pre-training in other domains). The framework, the generated CyLLMs, and the data are publicly available for use in cybersecurity applications.

1 INTRODUCTION

Large Language Models (LLMs) are transformer-based models (Vaswani et al., 2017) with billions of parameters, enabling extraordinary language understanding capabilities. Central to their development is the pre-training phase, where these models are exposed to vast amounts of unstructured text. This critical step infuses the models with extensive knowledge, allowing them to understand and generate human-like text effectively. The outcome of this pre-training process is the creation of base or foundation LLMs. The foundation models can be further fine-tuned for language problem-solving (e.g., text classification, Q&A, summarization, etc.), often with significantly fewer data and computer resources. The result of such fine-tuning process is the instruct LLMs.

Commercial LLMs (e.g., OpenAI's GPTs (Brown et al., 2020)) are general-purpose instruct models with strong task-solving capacity. However, commercial LLMs may involve data transmission to third-party servers, raising privacy and security concerns in sensitive industries and businesses. More recently, high-tech companies (e.g., Meta, Mistral) have published highly capable open-source LLMs that are competitive with commercial LLMs. With a high level of customization and data transparency, developing private LLMs based on these open-source models has become a new trend.

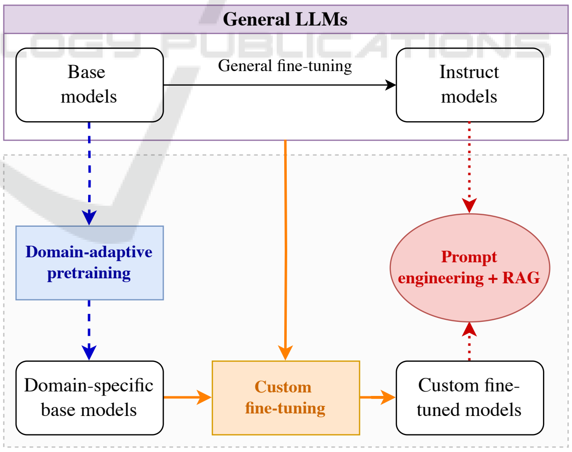


Figure 1: Main methods for utilizing LLMs in downstream tasks.

Users can choose to utilize three main methods to utilize open-source LLMs, as shown in Figure 1.

1. **Domain-Adaptive Pre-Training** (as shown by

Mai, K., Beuran, R. and Inoue, N.
CyLLM-DAP: Cybersecurity Domain-Adaptive Pre-Training Framework of Large Language Models.
DOI: 10.5220/0013094800003899
In *Proceedings of the 11th International Conference on Information Systems Security and Privacy (ICISSP 2025) - Volume 2*, pages 24-35
ISBN: 978-989-758-735-1; ISSN: 2184-4356

the blue box and dashed lines in Figure 1): To optimize the model's performance in a particular domain, users can employ domain-adaptive pre-training of the base models on unstructured text from the relevant knowledge domain. Domain-adaptive pre-training is a type of unsupervised training because the training data does not come with explicit annotations or labels. This approach results in domain-specific base models.

2. **Custom Fine-Tuning** (presented with orange box and lines in Figure 1): Custom fine-tuning utilizes annotated (supervised) datasets to adapt the model's capabilities for more effective performance in specific tasks. Custom fine-tuning can be implemented for general and domain-specific models, resulting in custom fine-tuned models.
3. **Prompt Engineering** (shown with red ellipse and dotted lines in Figure 1): Users design questions in a specific format (prompts) to interact with fine-tuned (instruct) LLMs. To ensure that LLMs are provided with the correct context, Retrieval-Augmented Generation (RAG) can be used to retrieve relevant documents for the question and incorporate these documents into the prompt.

In the figure presented, domain-adaptive pre-training, custom fine-tuning, and prompt engineering with RAG are not mutually exclusive. They can be combined to improve the LLM-based application's overall performance.

LLMs have been utilized in various cybersecurity sub-domains, marking a new branch of cybersecurity research, namely LLM for cybersecurity (LLM4Security) (Zhang et al., 2024). However, the domain specialization of LLMs in cybersecurity is usually under-discussed for different reasons. First, the cybersecurity domain is enormous and diverse, with vague boundaries from other domains. This prevents an instant solution from being able to collect and preprocess the data. To the best of our knowledge, no cybersecurity datasets are big enough for direct use in the LLM domain specialization. Second, the cost of implementing a solution to collect and preprocess data is significant when compared with the cost of RAG or fine-tuning.

This paper introduces a simple but effective framework based on domain-adaptive pre-training, named CyLLM-DAP, to ease the cybersecurity specialization process of LLMs. Using the framework, cybersecurity-specific LLMs (termed CyLLMs) are created by exposing two open-source LLMs (Llama 3 and Mistral v0.3) with 30 GB of cybersecurity text from heterogeneous sources. These CyLLMs are evaluated in two experiments, showing their capacity to improve fine-tuning effectiveness when compared to baseline LLMs. In our GitHub repository[1], we discuss the usage of the published framework, models, and training data. Note that we are actively updating the framework and related assets.

The remainder of the paper is structured as follows. In Section 2, we present background concepts together with relevant works. Section 3 describes in detail the framework architecture and implementation. We then discuss the use of the proposed framework for cybersecurity specialization of LLM in Section 4. The experiment to evaluate the CyLLMs is mentioned in Section 5. The paper ends with a conclusion and references.

2 BACKGROUND AND RELATED WORKS

In this section, we first introduce large language models (LLMs) and the backbone architecture used in most LLMs: transformers. We then present the concept of domain-adaptive pre-training. Finally, we discuss how LLMs are used in the cybersecurity domain.

2.1 Large Language Models

In 2017, Google researchers introduced transformers (Vaswani et al., 2017) with a multi-head attention mechanism, marking a new era in deep learning (DL) research. The unique attention mechanism allows transformers to capture dependencies and patterns in the input parallelly. Additionally, the effectiveness of transformers is based on transfer learning, in which models reuse knowledge they have already acquired and apply it to a new but related problem. For example, pre-training with textual financial data can improve the model's performance in solving finance-related problems. There are generally three main transformer architectures: sequence-to-sequence (encoder-decoder) models, decoder-only models, and encoder-only models. We mainly discuss decoder-only and encoder-only models, which are the most successful transformer-based architectures.

Encoder-only models, such as BERT (Devlin et al., 2019), work as an embedding module, taking the input sequence and output its representative vector. When adapting to downstream tasks, different types of layers, such as the softmax layer for classification tasks, are added on top of the encoder models to generate suitable answers (e.g., class labels).

Decoder-only models (GPT models) focus solely on the decoder component of the original transformer

[1] https://github.com/cyb3rlab/CyLLM-DAP

architecture. These GPT models (Radford et al., 2018) come in different sizes, ranging from millions to billions of parameters. GPT models with billions of parameters focusing on language understanding are called Large Language Models (LLMs).

The next token prediction is a core mechanism used in training LLMs. In this mechanism, the model learns to predict the next word (or token) in a sequence based on the context provided by the preceding words. In the training phase, when a sequence of k tokens $(s_1, s_2, ..., s_k)$ is inputted, the model learns to maximize the probability $\sum_i^k \log P_\Theta(s_i|s_1, s_2, ..., s_{i-1})$ with Θ is the model's parameters. The model's parameters are then updated, allowing it to minimize the prediction error between its output token and the expected one. This process iterates over vast amounts of text data, allowing the model to learn natural language's statistical properties and patterns.

The recent rise in popularity of open-source language model frameworks, as opposed to commercial ones like GPT-4, is attributed to their long-term advantages. Open-source frameworks offer greater customization, allowing users to adjust model parameters completely or partially to suit their training requirements. They also enable users to train models from scratch for experimentation or personal use. Furthermore, the increasing availability of cheaper and higher-performance computers makes training and hosting models in production more cost-effective. Privacy concerns also drive the preference for open-source frameworks, as companies and organizations aim to maintain control over their data. Recently, the Llama series (Touvron et al., 2023) and Mistral series (Jiang et al., 2023) are among the best open-source LLMs, comparable in performance with closed-source LLMs, such as GPT4.

2.2 Domain-Adaptive Pre-Training

The published LLMs are originally designed for general-purpose problem-solving. They struggle to perform well when tasked with understanding specialized knowledge domains. This is because LLMs are fundamentally statistical models that learn patterns from text data after exposure to extensive amounts of text. The insufficient expertise domain data in the pre-training stage reduces LLM's performance in the target domain.

Domain-adaptive pre-training, or continual pre-training, is an advanced technique to customize a general-purpose language model for a specific domain by continuing its training using the domain's data. This process aims to improve the model's performance in a particular field without starting from scratch. As shown in the survey (Ling et al., 2023) regarding the LLM domain specialization, this technique has been extensively applied in various domains (e.g., finance, law) to improve the LLM performance in such domains. For example, Wu et al. (Wu et al., 2023) develop PMC-LLaMa by pre-training the base model (Llama 2) with biomedical papers and books and subsequently fine-tuning for following instructions in the medical domain. In the evaluation of this research, domain-adaptive pre-training allows the model to reach 2.94% of the performance gain.

In cybersecurity, domain-adaptive pre-training is not a new concept. Before the era of LLMs, we already have various cybersecurity-specific encoder-only models, such as SecureBERT (Aghaei et al., 2023), CyBERT (Ranade et al., 2021) and CySecBERT (Bayer et al., 2024). To our knowledge, no public framework currently supports this domain-adaptive pre-training task for LLMs in cybersecurity. Table 1 shows the advancements of CyLLM-DAP over similar frameworks/models for the cybersecurity domain specialization pre-training process.

2.3 LLM Applications in Cybersecurity

Text data is a redundant and important source of information in many domains. In cybersecurity, text data can be obtained from Cyber Threat Intelligence (CTI) reports, emails, system logs, network design, security guidelines, etc. Text analysis and generation are unavoidable tasks in many cybersecurity solutions. At the simplest level, generating a summary report to explain the system's output is beneficial.

LLMs have been utilized in cybersecurity for different purposes. For example, LLMs can be used in various scenarios and training roles in cybersecurity training and education. Greco et al. (Greco et al., 2024) presents various promising strategies utilizing LLMs for PETA (Phishing Education, Training, and Awareness). LocalIntel (Mitra et al., 2024) is a LLM-based framework for generating organizational threat intelligence. Moreover, LLM can also be used for software vulnerability detection (Ferrag et al., 2024), malware dynamic analysis (Yan et al., 2023), hardware security and policy generation (Tarek et al., 2024), etc.

In a recent study (Zhang et al., 2023), various methods in the field are compared for addressing potential vulnerabilities in software systems. The study demonstrates that the use of transformers with pre-trained knowledge outperformed traditional machine-learning methods in the context of software vulnerability repair. Moreover, after pre-training these models on extensive codebase data, there is a notable 9.4%

Table 1: Comparision between CyLMM-DAP and other domain-adaptive frameworks/models in cybersecurity.

Frameworks/ Models	**Model Type**	**Dataset Publication**	**Model Publication**	**Data Collecting Scripts**	**Filtering and Preprocessing Scripts**	**Domain-Adaptive Pre-training Scripts**	**Framework Publication**
SecureBert	encoder-only		✓			✓	
CyBERT	encoder-only		✓				
CySecBert	encoder-only	✓	✓	✓	✓		
CyLLM-DAP	decoder-only	✓	✓	✓	✓	✓	✓

increase in accuracy. The authors suggest that pre-training the models with knowledge closely aligned with the target downstream task is a promising approach for enhancing their performance.

In a comprehensive survey conducted by Zhang et al. (Zhang et al., 2024) on the use of LLMs in cybersecurity, more than 180 projects across over ten downstream tasks are analyzed. The survey explicitly referenced only one paper (Jiang et al., 2024) that integrated domain-adaptive pre-training into the proposed approach for binary code analysis. This highlights the current lack of discussion on domain-adaptive pre-training for LLMs in cybersecurity applications.

3 CyLLM-DAP ARCHITECTURE

This section presents the general architecture of CyLLM-DAP. As shown in Figure 2, the framework contains six main components to support the domain-adaptive process in cybersecurity, including:

1. **Data Collection:** To collect data from different sources.
2. **Relevance Filtering:** To filter out irrelevant documents from the dataset.
3. **Quality Filtering:** To ensure the quality of the data, various methods can be utilized to filter out bad documents from the dataset.
4. **Data Anonymization:** To protect individuals' private and sensitive information.
5. **Data Deduplication:** To ensure the uniqueness of each document.
6. **Training**: To provide training scripts for the domain-adaptive pre-training process.

The framework provides a high level of customization by following object-oriented programming. To work with the framework, users can choose a default workflow or create a personal workflow of components to meet their needs. In addition to this, they can also implement their own components, using the inheritance mechanism in object-oriented programming.

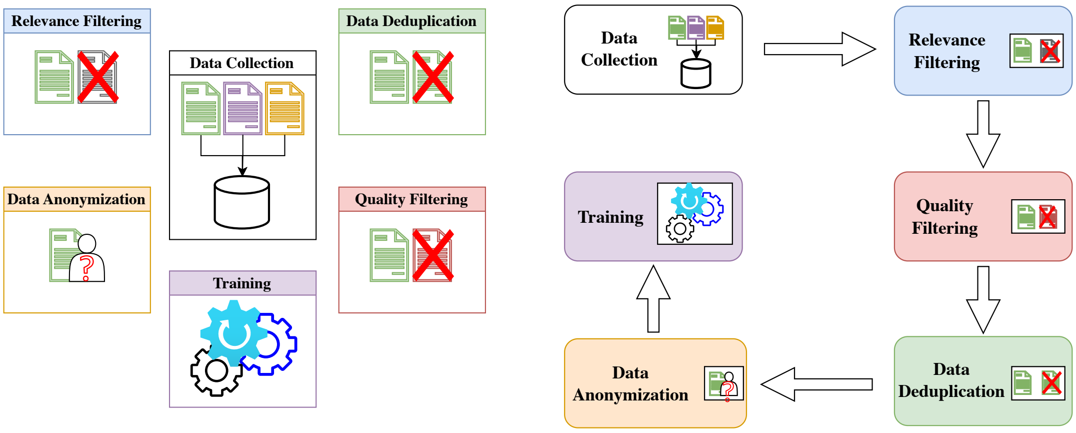
Note that not all of the framework's components are implemented from scratch by us. In such components, we reuse the best effort from other researchers and developers. We clearly mention this information when providing more details for each component in the following sections.

3.1 Data Collection

To ensure the generability of LLMs, training data should be collected from multiple data sources. The diversity of data exposes the model to a wide range of language patterns, ensuring fairness in the language capacity of the models across different scenarios. Different data sources are currently supported by CyLLM-DAP, including:

1. Web Data: Web data is text data obtained from web pages. This is the most common and redundant type of data. The Common Crawl dataset (Patel, 2020) is a web dataset that is crawled from the internet. The Common Crawl organization maintains this dataset by conducting regular scrawls, which started in 2007. Currently, Common Crawl is the biggest dataset with hundreds of TiB of data, spanning over billions of web pages. When working with this type of dataset, users can choose to work with the HTML representation (WARC format) or plain text (WET format) extracted from those web pages.

   Our framework supports the data collection using both of the mentioned data formats. However, the text extraction mechanism originally implemented by Common Crawl to create WET data is not optimal, as discussed in (Li et al., 2024). Instead, we use the trafilatura Python library (Barbaresi, 2021) to extract high-quality text from the raw HTML data (WARC format).
2. Academic Papers: Academic papers are another great source of data, providing high-quality text written by researchers. Currently, S2OCR (Lo et al., 2020) is the largest dataset of English academic papers that are well-established and maintained. Any papers contained inside S2OCR have already been transformed from different data formats (e.g., PDF, latex source code) into a user-friendly data type. Besides metadata such as corpus ID, the paper content is annotated with labels for easy extraction of titles, abstracts, and para-



(a) Framework components. (b) Example workflow.

Figure 2: CyLLM-DAP's components and example workflow in practical use.

graphs. Currently, CyLLM-DAP only supports this dataset for collecting academic papers.

3. Hugging Face Hub: Hugging Face (HF) (Delangue, 2024) is a famous machine-learning framework containing open-source libraries and tools for developing and deploying advanced natural language processing (NLP) models. In addition, the HF hub is a platform where researchers and developers can share or access a significant number of developed models and datasets. Therefore, the hub is an important data source in terms of redundancy, diversity, and data quality. In CyLLM-DAP, we currently support the data collection from Wikipedia and RedPajama (TogetherAI, 2023). Users can quickly adapt the framework for other datasets hosted in the HF hub.
4. Books: Books are another high-quality source of data that should be included when pre-training LLMs. While CyLLM-DAP currently supports the download of books from online sources, users should take care of the copyright problems when downloading them.
5. Code: Pre-training LLMs with code data has been proven effective for downstream tasks relevant to programming code. Code data can be collected from online code hosting or question-and-answer platforms (e.g., GitHub, StackOverflow).

CyLLM-DAP implements various collectors to support data collection from the mentioned sources and transform them into text-based data. Note that data collectors in CyLLM-DAP are implemented in streamlined modes to avoid downloading all the raw data into local storage. For this reason, collectors come with simple forms of relevance filters (discussed in 3.2.1) to remove irrelevant documents on the fly.

3.2 Data Filtering

In this section, we discuss two main components of CyLLM-DAP for data filtering: relevance filtering and quality filtering. Generally, a data filter will take a document as an input and output a boolean value indicating if the document meets some predefined requirements. Data filters can be used as part of the data collectors for preliminary filtering or as separate modules in the workflow.

3.2.1 Relevance Filtering

When preparing data for LLM's domain specialization, it is crucial to determine the relevance of data elements, such as documents, to the target knowledge domain. Depending on the characteristics of the target domain, different strategies can be recruited appropriately. For relevance filtering in the cybersecurity domain, as briefly mentioned earlier, the following characteristics should be considered:

1. Vague Domain Boundary: The cybersecurity domain does not have a clear boundary separating itself from other domains. Computer systems are deployed in many domains to automate their operations with high accuracy. However, these computer-based solutions also come with risks and cybersecurity challenges. To effectively address cybersecurity issues in such systems, it is necessary to have cross-domain knowledge. For instance, preventing online fraud in the internet banking system requires a combination of cybersecurity and financial expertise.
2. Broad and Diversity: Cybersecurity encompasses various sub-domains, including hardware security, software security, data security, and code security. The specificity of these sub-domains

varies, with some being quite broad, like CTI, and others containing more specialized or niche knowledge. For example, Generative AI Security is a special cybersecurity sub-domain, pertaining to security issues within generative AI-based tools and solutions, such as LLMs.

In CyLLM-DAP, there are two main methods used for relevance filtering, as follows.

Keyword-Based Relevance Filtering. The keyword-based approach begins with a list of cybersecurity keywords, regex, and URL patterns. CyLLM-DAP currently uses a list of 500 cybersecurity keywords (e.g., data security, data safety) and regex patterns (e.g., CVE). For URL patterns, we determine a list of 300 cybersecurity newspapers and blogs (e.g., www.scmagazine.com). Note that these lists are being updated frequently.

The keywords and patterns used in CyLLM-DAP are enclosed in filters, each of which has a different method for assessing various components (such as URLs, text, and titles) of the document for relevance filtering. One important advantage of keyword-based filtering is its running speed and multiprocessing friendliness. Each filter requires little memory to run the function. For this reason, we recommend using these keyword-based filters as the preliminary approach to handling big data sources.

Model-Based Relevance Filtering. In model-based filtering, language models are used to determine a document's relevance to the cybersecurity domain. A DL model is trained to classify documents into cybersecurity and non-cybersecurity categories. For this purpose, an encoder-only model, already pre-trained in the context of cybersecurity, is fine-tuned with a labeled dataset. After the fine-tuning, the model can provide a cybersecurity relevance score for an input document. With a threshold value of 0.5, any document with a cybersecurity relevance score under 0.5 will be removed from the dataset.

3.2.2 Quality Filtering

In LLM development, data quality is one of the most important factor to ensure the model's performance. From different LLM projects, various quality metrics are invented based on the observation of actual data. These quality signals/metrics are mostly calculated by analyzing the text's statistics, such as the ratio between the number of sentences starting with bullet points and other sentences.

In CyLLM-DAP, inspired by other LLM works (TogetherAI, 2023), we develop functions for quality metrics and rules. The metrics functions take a document as input and output relevant metrics score. Additionally, rules in CyLLM-DAP are implemented as filters, in which some thresholds are applied to the document's quality metrics to determine if it is a good document. For example, the mentioned ratio should not exceed 0.7 to indicate a good document.

While these metrics are proven effective, they are not universal and should be choose wisely depending on the target tasks. Beside the list of default metrics and rules for cybersecurity domain, LLM developers can design their own functions to meet their needs.

3.3 Data Deduplication

Duplicate entries in training datasets can result in over-fitting and create a false impression of improved performance during training. Recent research, as highlighted in (Lee et al., 2022), underscores the importance of data deduplication, emphasizing its potential to enhance model performance in a more balanced and comprehensive manner. To deduplicate data in general, we first need to identify duplicates using similarity search and remove them.

In an exhaustive similarity search, each pair of documents in the dataset is usually compared to determine their similarity. This mechanism imposes a too-high time cost and is not suitable for handling large amounts of data. In CyLLM-DAP, we use a popular method for efficient similarity search, namely LSH (Locality Sensitive Hashing) (Dasgupta et al., 2011). In general, LSH aims to reduce the data dimensions while maintaining local distances between data points. The output of this method is buckets of similar documents. Subsequently, only the longest document from each bucket is reserved, while others are removed from the dataset.

When deduplicating small datasets, the whole process can be done in the computer's RAM. However, a large amount of intermediate data (e.g., dense hash signatures) is generated during the run of LSH. For this reason, CyLLM-DAP splits the deduplicating process into smaller steps and stores any intermediate data in local storage. This allows CyLLM-DAP to work with large datasets.

3.4 Data Anonymization

Data anonymization is a data sanitizing technique that protects the individual's privacy or sensitive information. In this technique, personally identifiable information (PII) is detected and removed (or modified) from the text before feeding to LLMs. As a result, LLMs do not remember personal information, avoiding being gathered by cyber attackers in the inference stage.

CyLLM-DAP utilizes presidio anonymizer (Microsoft, 2024) developed by Microsoft to implement its anonymizing function. Currently, only four types of PII are considered, including email, URL, phone number, and credit card. More PII will be added in the future. Additionally, users can create customized anonymizing functions and integrate them into CyLLM-DAP.

3.5 Training

In CyLLM-DAP, we create various scripts that users can directly use for domain specialization. These scripts are mostly based on the HuggingFace library and its code examples. Since HuggingFace provides leading libraries and tools for working with Machine Learning and LLM models, using their examples ensures a robust yet simple approach to LLM development.

The provided scripts follow two common LLM pre-training paradigms, considering the available computer resources. One is the full pre-training, in which all model parameters will be alternated during the pre-training process. This approach requires a large amount of GPU RAM with a possibility of catastrophic forgetting of old knowledge. The other is to partially update the model's parameters, which is more suitable for a low-resource computing environment.

4 CyLLMs CREATION USING CyLLM-DAP

In this section, we discuss the development process of cybersecurity-specific foundation LLMs (CyLLMs). Two CyLLM versions (CyLlama3 and CyMistral) are based on two corresponding open-source LLMs: Llama-8B-v3 and Mistral-7B-v0.3. We first present the data preparation process. Then, we discuss the domain-adaptive pre-training process to specialize the foundation models in cybersecurity knowledge. Both of these tasks are implemented using CyLLM-DAP. Lastly, we examine the impact of domain specialization through experiments on two distinct downstream tasks. The whole process is demonstrated in Figure 3, in which the data preparation and domain-adaptive pre-training process are discussed in Section 4.1 and 4.2, respectively. Subsequently, the experiment is presented in Section 5.

4.1 Data Preparation

The training data used in the specialization process of LLM are collected via CyLLM-DAP using the default setting. Table 2 shows the data sources which we collect data from. The data collection process is conducted over six months (from Jan. to Jun. 2024) using four computing nodes with high-speed internet. As we can see in Table 2, there is no code data included since we are not focusing on solving code-related downstream tasks. We plan to include code data for code-related downstream tasks in our future work.

Table 2: Cybersecurity text data for domain-adaptive pre-training.

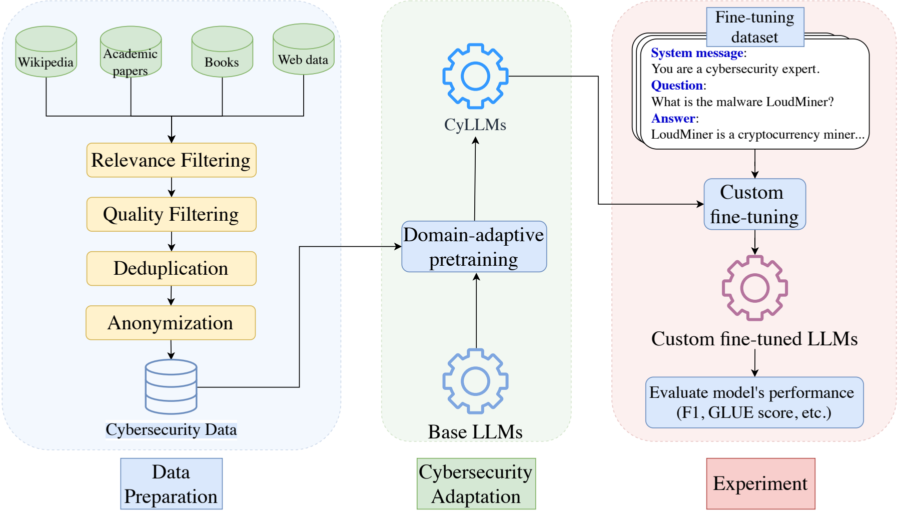
Data sources	**Size**	**Original Sources**
Wikipedia	∼500 MB	Wikipedia dataset on HuggingFace hub
Academic papers	∼3 GB	S2OCR
Books	∼200 MB	Online book libraries
Web data	∼26 GB	Common Crawl RedPajama
Total	∼30 GB	

Using CyLLM-DAP, we apply various data filters to the dataset to ensure its quality and relevance. These include:

1. Language filtering: In this dataset, only text written in the English language is kept.
2. Relevance filtering: In this step, the preliminary relevance filters (with keywords and patterns) are applied during the data collection. Model-based filtering is also used as a separate module. Regarding academic papers, we use the API search function provided by S2OCR authors to obtain pertinent documents related to a list of cybersecurity keywords.
3. Quality filtering: All the default metrics and rules implemented by CyLLM-DAP for the cybersecurity domain are used. Note that we also filter out toxic documents at this step.
4. Deduplication: We apply the deduplication process on the dataset to remove duplicated documents.
5. Anonymization: The default anonymization function of CyLLM-DAP is used.

4.2 Domain-Adaptive Pre-Training

As mentioned in Section 3.2.1, the cybersecurity domain does not have a clear boundary with other



System message:
You are a cybersecurity expert.
Question:
What is the malware LoudMiner?
Answer:
LoudMiner is a cryptocurrency miner...

Figure 3: The development process of CyLLMs with CyLLM-DAP.

knowledge domains. Depending on the application, solving cybersecurity downstream tasks may require knowledge from other domains. Specializing the models in the cybersecurity domain should not cause knowledge loss in other domains. For this reason, we avoid fully training the model in which all of their parameters are updated. Instead, a small amount of parameters is set to be altered during the continual pre-training process using LORA (Low-rank Adaptation) (Hu et al., 2021). LORA is one of the most popular methods for working with LLMs, aiming to enable LLM training in low-resource settings. LORA works by freezing the model's weights and introducing smaller trainable matrices of parameters. By this, only a few weights are updated to adapt the model with new information.

Since most model parameters are frozen during the domain-adaptive pre-training process, catastrophic forgetting is alleviated. This preserves the model's pre-trained knowledge relevant to other domains. In LORA training, rank and alpha are two important hyperparameters controlling the size of the trainable matrix. We set the rank and alpha hyperparameters (primary hyperparameters used in LORA) to be high (64 and 128, respectively) to effectively inject new knowledge into the models.

The domain-adaptive pre-training process uses H100 GPU computers, with 123 hours for training CyLlama3 and 135 hours for CyMistral for one epoch. Our approach involves utilizing a context length of 1024 and a batch size of 64 without applying quantization to the model. Quantization is a method used to reduce GPU memory consumption during LLM pre-training by mapping high-precision values to lower precision. While this approach can effectively reduce GPU memory usage, it can also impact the quality of the pre-training process. Since our models can be accommodated within GPU memory during training with LORA, we opt not to employ quantization in order to maintain the quality of the pre-training process.

The result of the domain-adaptive pre-training process is two cybersecurity-specific foundation LLMs, namely CyLLMs. These CyLLMs come in two versions, each based on its respective foundation model. To elaborate, CyLlama3 is the cybersecurity-specific model of Llama-8B-v3, while CyMistral3 is built on Mistral-7B-v0.3. In the following section, we will carry out experiments on these CyLLMs to assess their performance on downstream tasks.

5 EVALUATION

In this section, we employ two downstream tasks to validate the effectiveness of CyLLMs (generated using CyLLM-DAP) in various cybersecurity applications, particularly in tasks involving custom datasets. Several important considerations are as follows:

1. In this evaluation, our focus is not on creating multi-purpose instruct LLMs that can excel in every cybersecurity task. Instead, we conduct two experiments where relevant models are fine-tuned separately.
2. The evaluation wants to confirm the performance of these CyLLMs in private and custom datasets. This is a common use case for open-source LLMs in the field of cybersecurity. Therefore, utilizing private datasets that have not been publicly disclosed is more appropriate for this purpose.

Table 3: The list of LLMs involved in the experiments and the type of training method already applied to them.

Model Group	General Pre-training	Domain-adaptive Pre-training	General Fine-tuning
Group **B**: - **B**aseMistral - **B**aseLlama3	✓		
Group **I**: - **I**nstructMistral - **I**nstructLlama3	✓		✓
Group **C**: - **C**yMistral - **C**yLlama3	✓	✓	

As shown in Table 3, there are three groups of LLMs related to the experiments. Each group contains corresponding models from the Llama 3 and Mistral v0.3 LLM families.

- Group B includes the base (foundation) models Llama-3-8B (BaseLlama3) and Mistral-7B-v0.3 (BaseMistral) originating from the company (e.g., Meta, Mistral). Base models are developed via general pre-training.
- Group I includes Llama-3-8B-Instruct (InstructLlama3) and Mistral-7B-Instruct-v0.3 (InstructMistral) that the technology company originally published. These models have undergone both general pre-training and general fine-tuning.
- Group C comprises cybersecurity-specific LLMs (CyLLMs), namely CyLlama3 and CyMistral. These are developed based on BaseLlama3 and BaseMistral, respectively.

To confirm the effectiveness of the domain-adaptive pre-training, all the models in groups B, I, and C are custom fine-tuned to solve the task. After the fine-tuning process, we calculate and compare their performances using metrics. The details for each experiment are presented in the following sections.

5.1 Task 1: Text Classification

This text classification experiment aims to assess LLM's ability to generate short responses to identify the category of a cybersecurity text among many classes. For this experiment, we create a dataset of 145,000 data samples. Each data sample contains a text relevant to one cyberattack technique labeled with its MITRE ATT&CK's technique ID. MITRE ATT&CK (Strom et al., 2018) is a globally accessible knowledge base that provides standardized knowledge to cybersecurity practitioners regarding attacking tactics, techniques, procedures and malicious entities such as attackers, campaigns, and tools. This task is a multi-class classification with 628 ATT&CK technique IDs as classes. In general, the dataset generation has two stages:

Table 4: The task-specific dataset used in the experiment with examples.

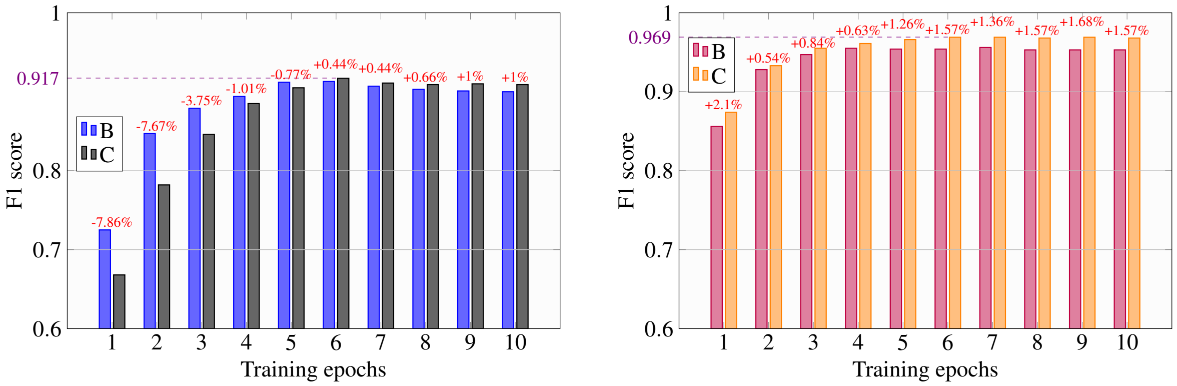
Related Task	Example
Text classification (145,000 samples)	- **System message**: You are a cybersecurity expert. Below is an instruction that describes a task in the cybersecurity domain, paired with an input that provides further context. Write a response that appropriately completes the request. - **Instruction**: You are given a text description of a procedure example. Identify the MITRE ATT&CK technique used. - **Input**: Siloscape leverages a sophisticated form of API call concealment... - **Output**: Obfuscated Files or Information T1027
Question & Answering (22,000 samples)	- **System message**: You are a helpful cybersecurity expert. - **Question**: What is the malware LoudMiner? - **Answer**: LoudMiner is a cryptocurrency miner that uses virtualization software to siphon system resources...

1. Stage 1: We utilize an automatic report analysis framework, namely RAF-AG (Mai et al., 2025), to initiate the data-generating process. Using the framework, the input CTI reports are transformed into corresponding cyberattack paths. For each attack path, the attacking technique ID of an attack step, along with its correlated text, is gathered.
2. Stage 2: We use OpenAI GPT 3.5, a powerful closed-source LLM, for data augmentation, a process by which an input text is paraphrased into various text patterns.

We utilize the Alpaca prompt template (Taori et al., 2023) to format the data into a single prompt. Because instruct models (group I) are originally aligned with different prompt templates during its general fine-tuning, only B and C models are involved in this experiment. After the custom fine-tuning, the models are expected to recognize the most probable technique ID for the input text.

During fine-tuning, a single A100 40 GB computer is used with LORA (rank of 64 and alpha of 128) and a learning rate of $1e-4$. 90% of the dataset is allocated for fine-tuning, while the remaining portion is reserved for evaluation. Model performance is documented at the end of each epoch over a span of ten epochs. During the evaluation phase, we gather LLM's responses for all data samples in the evaluation set. Subsequently, we extract technique IDs from these responses and compare them with the expected ones to calculate the F1 score.

As we can see from Figure 4, introducing cybersecurity knowledge to the LLMs results in overall performance improvements compared to the baseline foundation models (B models) (see values highlighted



(a) Llama 3 series.

(b) Mistral v0.3 series.

Figure 4: The performance of C models (CyLLM) and B models (Base models) measured via the F1 score for text classification. The performance differences in percentage between C models and B models are shown in red color. Horizontal violet dashed lines and numbers show the maximum F1 scores in each chart. The left-hand side chart shows the Llama 3 LLM series, and the right-hand side chart shows the Mistral v0.3 LLM series.

in red). This impact is consistently observed in every fine-tuning epoch with Mistral v0.3 models (the right chart). Conversely, the effectiveness of cybersecurity knowledge varies across epochs for Llama 3 models. This effectiveness becomes more apparent when the number of epochs exceeds six (≥ 6). Furthermore, models from the Mistral v0.3 series (shown in the right chart) demonstrate superior performance, with a maximum F1 score of 0.969 compared to 0.917 for the Llama 3 series (displayed in the left chart).

5.2 Task 2: Question & Answer

In this experiment, we develop a simple Q&A chatbot that can accurately respond to user inquiries regarding CTI, such as the attacking techniques, malware, attackers. The experiment aim to evaluate LLM's ability to generate long responses, with a stringent requirement for the LLMs to adhere closely to the expected answers. This is particularly crucial in situations where the chatbot needs to provide accurate and comprehensive information without any fabrication.

We utilize a dataset of 22,000 conversation samples, encompassing questions and corresponding expected answers. Additionally, the dataset features a system message prompting LLM to assume the role of a cybersecurity expert during the interaction. A data sample can be seen in Table 4.

All the models in Table 3 are involved in this experiment. They are subsequently fine-tuned with the dataset to acquire the Q&A capacity. Note that we employ the default chat templates originally created by the companies behind each model family. In evaluating the text generated by the fine-tuned LLMs, we take into account the following metrics:

1. **GLEU (Google BLEU) score** (Wu et al., 2016) is commonly used to evaluate the performance of translators by measuring the similarity between machine-translated text and the expected output.
2. The **BERTScore** (Zhang et al., 2020) leverages contextual embeddings generated by the BERT (Devlin et al., 2019) model. This metric is widely used for assessing the similarity between texts and is proven to produce results similar to human evaluations.
3. A high-quality commercial LLM (GPT-4o) will act as a judge (**LLM-as-a-judge**) to evaluate the generated response in terms of Understandability (clarity of the response), Relevance (adequacy and pertinence of the information), Naturalness (human-like quality of the response), and Hallucination (incorporation of incorrect information).

Table 5 shows the performance of related models for Q&A fine-tuning regarding six main metrics: GLEU score, BERTScore, Understandability, Naturalness, Relevance, and Hallucination. Word count is a sub-metric mainly used for reference. From this table, we can observe that:

- In general, incorporating cybersecurity knowledge into the models enhances their performance compared to models equipped only with general knowledge. CyMistral stands out as the top-performing model (in Mistral v0.3 series) across all six metrics, while CyLlama3 outperforms others in 4 out of 6 metrics within the Llama 3 models. When comparing the best with the second-best model in each series, integrating cybersecurity knowledge leads to performance improvements of 4.75% (GLEU score) for the Llama series, and 3.07% (GLEU score) and 2.85% (BERTScore) for the Mistral series. These perfor-

Table 5: The performance of the created LLMs for the Q&A task. The underlined scores show the best models within the same family (same background color). The bold scores show the best model for specific metrics among all of the involved models. Unlike other metrics, models with less hallucination score is better.

Model	GLEU score	BERTScore	Understandability	Naturalness	Relevance	Hallucination	Average Length
BaseLlama3	0.569	0.718	0.846	0.77	0.618	0.565	115.615
BaseMistral	0.778	0.91	0.957	0.937	0.835	0.119	64.699
InstructLlama3	0.565	0.707	0.828	0.747	0.603	0.592	116.474
InstructMistral	0.781	0.911	0.954	0.934	0.828	0.118	64.01
CyLlama 3	0.596	0.717	0.854	0.762	0.63	0.563	111.943
CyMistral	**0.805**	**0.937**	**0.959**	**0.939**	**0.84**	**0.103**	64.022

mance gains are comparable with those acquired by injecting domain-specific knowledge in other domain areas (e.g., medical (Wu et al., 2023)).

- Comparing model families, Mistral models generally outperform Llama 3 counterparts, particularly in terms of the GLEU score. The leading Mistral model, CyMistral, achieves a GLEU score of 0.805, whereas the best Llama3 model, CyLlama3, scores 0.596. This performance difference can be attributed to Llama3 models generating longer responses compared to Mistral models (as indicated in the word count column). The discrepancy in length between the generated text and the expected text significantly impacts the computation of the GLEU score, resulting in a lower output score for Llama 3 models.
- It is evident that incorporating cybersecurity knowledge does not result in a significant performance improvement for the Understandability and Naturalness metrics. This is primarily because all the models involved are based on foundational models pre-trained with high-quality text to generate coherent and natural language. Additionally, Llama 3 models cannot capture all the necessary information present in the expected answer as effectively as the Mistral models (as indicated in the Relevance metric). In terms of the Hallucination metric, the Llama 3 models tend to generate supplementary information, which may be inaccurate, thereby resulting in a high Hallucination score as judged by the LLM.

Generally, LLM judgment is beneficial and trustful because the findings align with other deterministic metrics like BERTScore and GLEU score. Furthermore, it can be inferred that domain-adaptive pre-training for Llama 3 models is not as effective as for Mistral models. This indicates the need for additional considerations when utilizing Llama 3 models.

6 CONCLUSION

This paper presented CyLLM-DAP, a framework developed to facilitate essential tasks in the cybersecurity specialization process for open-source large language models. The framework consists of modules that can be utilized as they are or customized to gather data and ensure data quality before conducting domain-adaptive pre-training.

We illustrated the use of CyLLM-DAP for creating domain-adaptive pre-training or baseline models during fine-tuning for the text classification task. Moreover, comparable to domain-adaptive pre-training in other domains, the LLM's cybersecurity specialization can yield important performance improvement of up to 4.75% (for Q&A task) when compared with the general base and instruct models.

In our future work, we plan to address this study's limitations by incorporating a diversity of model sizes, families, and data sources. The framework, models, and cybersecurity data are accessible to the public and will undergo regular updates to facilitate the integration of LLMs in the cybersecurity field.

REFERENCES

Aghaei, E., Niu, X., Shadid, W., and Al-Shaer, E. (2023). SecureBERT: A Domain-Specific Language Model for Cybersecurity.

Barbaresi, A. (2021). Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 122–131. Association for Computational Linguistics.

Bayer, M., Kuehn, P., Shanehsaz, R., and Reuter, C. (2024). CySecBERT: A domain-adapted language model for the cybersecurity domain. *ACM Trans. Priv. Secur.*, 27(2).

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., et al. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Dasgupta, A., Kumar, R., and Sarlós, T. (2011). Fast locality-sensitive hashing. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1073–1081.

Delangue, C. (2024). Hugging face. https://huggingface.co/.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North*, pages 4171–4186. Association for Computational Linguistics.

Ferrag, M. A., Battah, A., Tihanyi, N., Jain, R., Maimut, D., Alwahedi, F., et al. (2024). SecureFalcon: Are We There Yet in Automated Software Vulnerability Detection with LLMs?

Greco, F., Desolda, G., and Viganò, L. (2024). Supporting the Design of Phishing Education, Training and Awareness interventions: An LLM-based approach. In *2nd International Workshop on CyberSecurity Education for Industry and Academia*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., et al. (2023). Mistral 7B.

Jiang, N., Wang, C., Liu, K., Xu, X., Tan, L., and Zhang, X. (2024). Nova: Generative Language Models for Assembly Code with Hierarchical Attention and Contrastive Learning.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. (2022). Deduplicating Training Data Makes Language Models Better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445. Association for Computational Linguistics.

Li, J., Fang, A., Smyrnis, G., Ivgi, M., Jordan, M., et al. (2024). DataComp-LM: In search of the next generation of training sets for language models.

Ling, C., Zhao, X., Lu, J., Deng, C., Zheng, C., Wang, J., et al. (2023). Domain Specialization as the Key to Make Large Language Models Disruptive: A Comprehensive Survey.

Lo, K., Wang, L. L., Neumann, M., Kinney, R., and Weld, D. (2020). S2ORC: The Semantic Scholar Open Research Corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Mai, K., Lee, J., Beuran, R., Hotchi, R., Ooi, S. E., Kuroda, T., and Tan, Y. (2025). RAF-AG: Report analysis framework for attack path generation. *Computers & Security*, 148:104125.

Microsoft (2024). Microsoft Presidio - data protection and anonymization SDK. https://microsoft.github.io/presidio/.

Mitra, S., Neupane, S., Chakraborty, T., Mittal, S., et al. (2024). LOCALINTEL: Generating Organizational Threat Intelligence from Global and Local Cyber Knowledge.

Patel, J. M. (2020). *Introduction to Common Crawl Datasets*, pages 277–324. Apress.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training.

Ranade, P., Piplai, A., Joshi, A., and Finin, T. (2021). CyBERT: Contextualized embeddings for the cybersecurity domain. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3334–3342.

Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., and Thomas, C. B. (2018). Mitre Att&ck: Design and Philosophy.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. (2023). Stanford Alpaca: An Instruction-following LLaMA model.

Tarek, S., Saha, D., Saha, S. K., Tehranipoor, M., and Farahmandi, F. (2024). SoCureLLM: An LLM-driven Approach for Large-Scale System-on-Chip Security Verification and Policy Generation.

TogetherAI (2023). RedPajama: an open dataset for training large language models.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., et al. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., et al. (2017). Attention is All you Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, volume 30, pages 6000–6010. Curran Associates, Inc.

Wu, C., Lin, W., Zhang, X., Zhang, Y., Wang, Y., and Xie, W. (2023). PMC-LLaMA: Towards Building Open-source Language Models for Medicine.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., et al. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.

Yan, P., Tan, S., Wang, M., and Huang, J. (2023). Prompt Engineering-assisted Malware Dynamic Analysis Using GPT-4.

Zhang, J., Bu, H., Wen, H., Chen, Y., Li, L., and Zhu, H. (2024). When LLMs Meet Cybersecurity: A Systematic Literature Review.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.