DrLLM: Prompt-Enhanced Distributed Denial-of-Service Resistance Method with Large Language Models

1st Zhenyu Yin[1,3,*], 2nd Shang Liu[1,2,3], 3nd Guangyuan Xu[1,2,3]

[1]*Shenyang Institute of Computing Technology, Chinese Academy of Sciences*
[2]*University of Chinese Academy of Sciences*
[3]*Liaoning Key Laboratory of Domestic Industrial Control Platform Technology on Basic Hardware and Software*
congmy@163.com, {liushang221, xuguangyuan18}@mails.ucas.ac.cn

Abstract—The increasing number of Distributed Denial of Service (DDoS) attacks poses a major threat to the Internet, highlighting the importance of DDoS mitigation. Most existing approaches require complex training methods to learn data features, which increases the complexity and generality of the application. In this paper, we propose DrLLM, which aims to mine anomalous traffic information in zero-shot scenarios through Large Language Models (LLMs). To bridge the gap between DrLLM and existing approaches, we embed the global and local information of the traffic data into the reasoning paradigm and design three modules, namely Knowledge Embedding, Token Embedding, and Progressive Role Reasoning, for data representation and reasoning. In addition we explore the generalization of prompt engineering in the cybersecurity domain to improve the classification capability of DrLLM. Our ablation experiments demonstrate the applicability of DrLLM in zero-shot scenarios and further demonstrate the potential of LLMs in the network domains. The implementation of DrLLM is available at https://github.com/liuup/DrLLM.

Index Terms—Distributed Denial-of-Service, zero-shot, Large Language Models

I. Introduction

The Distributed Deniel-of-Service (DDoS) threats in cyber space are becoming increasingly sophisticated and complex, posing a threat to critical Internet devices and systems including routers, switches or firewalls. Today, the scale of the Internet and the Internet of Things (IoT) is much larger than before, and the demand for network security measures is also increasing [1]. Although the Instrusion Detection System (IDS) countermeasures against DDoS are developing rapidly, network attack events are still increasing. Imperva released their report [2], claiming the DNS attacks increased in number by 215% when comparing H1 2024 with the same period in 2023 and the total number of recorded DDoS attacks surged 111% compared to the previous year.

Over the past few years, we have witnessed Large Language Models (LLMs), such as ChatGPT [3], Llama [4], Deepseek [5] and the other models [6], [7] have make siginificant influence in artificial intelligence (AI), these LLMs include leading and fancy model architecture, and were trained from mega-billion tokens which compressed with domain knowledge. As generative models, it can not only generate a variety of creative responses to meet user needs, but it is also easy to use and only requires a simple prompt to drive it [8]. The fantastic capability has been proved on neural language processing (NLP), multimodality fusion and other fields [9] [10], which even laying the foundation stone for Artifical General Intelligence (AGI).

Inspired by the success of LLMs in many fields, especially textual information mining. In this paper, we introduce DDoS Resistance Large Language Model (DrLLM) which contains three modules, a novel method that can efficiently extract the vast network traffic knowledge contained in pre-trained LLMs via progressive Role Reasoning pipeline by prompting. With our prompting strategy, by aligning structed network streaming data into text space to preserve the full semantics, we can embed global information and local information in prompt template to enhance the classifiction capability. We evaluated the performance of DrLLM with the public dataset CICD-DoS2019 [11], and we also conducted ablation experiments on zero-shot scenarios to validate the functionality of DrLLM's core modules.

We present the following main research contributions:

- We propose a progressive DDoS traffic detection framework based on LLMs which is called DrLLM, which can efficiently classify network flow data and improve the interpretability of the classification basis.
- We innovatively use Knowledge Embedding to embed the global information of the data into prompting template, and conduct ablation experiments to demonstrate its effectiveness.
- We carefully crafted multiple prompting templates and Constrain-of-Deviation (CoD) and Chain-of-Thought (CoT) has been used in Token Embedding module to embed the local information and enhance the classification performance of LLMs.

II. BACKGROUND AND RELATED WORKS

1) Cyber IDS and Method: For DDoS traffic detection and classification tasks, existing IDSs are usually based on neural network architectures and the final results reached state-of-the-art (SOTA). Yisroel Mirsky et al. [12] presented Kitsune based on autoencoders to detect abnormal traffic data. Lucid [13] was a lightweight deep learning solution in recource-constrained environments. IDS-INT [14] used transformer-based transfer learning for imbalanced network traffic and detect minority attacks. In practical scenarios, they are mostly deployed on the data plane or control plane of a programmable switch [15], [16] to handle large-scale hybrid and dynamic traffic attacks, and differentiate between normal and abnormal patterns.

2) LLMs Enhanced Cybersecurity: Large Language Models (LLMs) can excel across various domains, including Neural Language Processing (NLP) and Computer Vision (CV) [3], [4], and recent research in time series forecasting [17], [18], information mining [19], [20] has demonstrated the effectiveness of pre-trained models that can be fine-tuned for various multimodality tasks [21]. In addition, some methods have been proposed and applied to optimize LLMs [8], [22]. Current research demonstrates the potential of leveraging LLMs in network security, it is not only used to detect software vulnerabilities [23], but also to detect DDoS traffic [24] and generate mitigation strategies [25] to safeguard against evolving cyber threats [26].

III. OVERVIEW AND DESIGN

DrLLM has two primary objectives: 1. Preserve the same semantic information in tabular data and text data, and establish a new paradigm to align tabular data to text modality. 2. Through specific feature engineering, relying on LLMs to mine network flow data, build progressive reasoning pipeline. In order to achieve these two goals, we designed three main modules for DrLLM: Knowledge Embedding, Token Embedding, and Role Reasoning. The overview of DrLLM architecture is shown in Fig 1.

A. Knowledge Embedding

We assign expert assumptions at the beginning of Knowledge Prompt to converge the reasoning ability of large models to the network field and clarify specific network traffic flow classification tasks. In the zero-shot scenario, for LLMs, the representation of a single data point alone may lack prior information for judgment, because this ignores the global data feature distribution of the dataset. After the data preprocessing, we calculate the 5-tuple characteristics G_m (maximum, minimum, median, mean and variance) for each column in dataset as the global information. This simple and effective method can serve as prior knowledge for LLMs and inform the global data distribution of the dataset in advance. To reduce the output length in this module and increase the efficiency of reasoning, we constrain its output at the end position of the prompt to reduce the output content of this step.

The detailed process of Knowledge Embedding is as follows: Assume that for the dataset $D_{n\times m}(X) =$ $[X_1, X_2, ..., X_n]^\top$, where $X_n = (x_1, x_2, ..., x_m)$ and features are $Features = (F_1, F_2, , ...F_m)$, for every column X_m, to calculate the maximum a, minimum b, median c, mean d and variance v, and finally get $Global_m = [G_1, G_2, ..., G_m]^\top$, where $G_j = (a_j, b_j, c_j, d_j, v_j)_{\{j=1,2,...,m\}}$. Then construct the string:

$$K(G) = Concat(Global_m \oplus S) \tag{1}$$

where $Concat$ means converting the elements in the vector into strings and concatenating them with $S =$ $("Max", "Min", "Median", "Mean", "Variance")^\top$. Finally we embed K into Knowledge Prompt (KP).

B. Token Embedding

In order to enable LLMs such as ChatGPT to understand network flow data, we designed a module called Token Embedding, which uses text to represent the specified information flow. We choose LLMs as our inference backbone, and we will introduce the design of the Token Embedding module below and describe in detail how Constrain-of-Deviation (CoD) and Zero-shot Chain-of-Thought(CoT) in Prompt Engineering improve and enhance the information extraction and classification accuracy of LLMs.

1) Constrain-of-Deviation (CoD): LLMs with decoder-only architecture are very suitable for generative tasks. They can flexibly generate text and code information, but their current structured content (such as XML or JSON) output capabilities are still weak [27]. When LLMs are integrated into our inference workflows, we need to contrain the outputs to follow specific standards. For the binary classification task of network traffic flow, in order to ensure consistency and predictability of output and reduce the hallucination, we created CoD Prompt to constrain the model to output according to instructions. This simple but very effective method allows us to use regular expressions to extract classification results from the output of LLMs.

2) Zero-shot Chain-of-Thought(CoT): Zero-shot-CoT [28]is a zero-shot template-based prompting for chain of thought reasoning. Different from few-shot prompting with examples, it can enlighten the LLMs to make progressive thinking and perform complex reasoning for each instruction in a simple but effective pattern. We use CoT to guide the LLMs to progressively analyze the various characteristic values of network flow data and help improve calculation accuracy.

3) Token Prompt: In the Token Prompt (TP), the specific network traffic task representation is initially indicated and the LLMs are directed to provide a probabilistic judgment based on each piece of data. We then construct Token Prompt based on CoD and Zero-shot-CoT, and use it as the benchmark input of the Token Embedding module. For tabular data of network flows, we treated as a new data modality and embed it into prompt to align it to the text modality, and then feed it into LLMs for reasoning. The specific process is as follows: Given a dataset $D_{n\times m}(X) = [X_1, X_2, ..., X_n]^\top$,
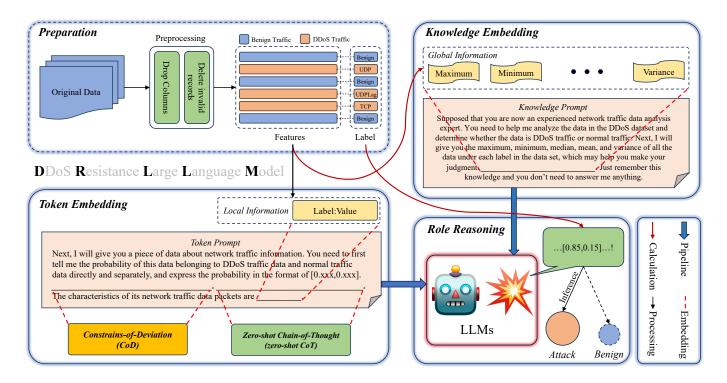
Fig. 1. The overall framework of DrLLM, we have divided the framework into four main modules: preparation module, knowledge embedding module, token embedding module and role reasoning module.

which owns $Feature_m = [F_1, F_2, ..., F_m]$, for every $X_n = [x_1, x_2, ..., x_m]$, we construct the text string $TP_i(F_j, X_i) = Concat(F_j : X_i)_{\{j=1,2,...,m,i=1,2,...,n\}}$ for each piece of data, and then embed TP_i into TP.

C. Role Reasoning

Based on Knowledge Embedding and Token Embedding, we build the progressively Role Reasoning pipeline on top of these two modules. In the first step, we embed the global information into the Knowledge Prompt KP, and feed KP as the first step input into the LLM to obtain the output R_i^1. In the second step, R_i^1 and Token Prompt TP_i are sequentially fed into the LLM for reasoning to obtain the output R_i^2. Finally, we use regular expressions to obtain the classification probability P_i from R_i^2 and calculate the evaluation result E_i with the true value $\hat{P}_i$ of the data X_i.

IV. EXPERIMENTAL EVALUATION

A. Datasets and Data Preprocessing

The DrLLM prototype is implemented with Python 3.11.9. We conducted extensive experiments for DrLLM on CICD-DoS2019 dataset [11]. The original dataset contains approximately 29GB of DDoS traffic data and benign traffic data, each with 88 features. According to the experimental needs, we preprocessed the data and deleted each row of data containing "NaN" or "Inf" values and we converted all DDoS traffic label to "Attack" and make benign traffic data unchanged.

B. LLM Backbones

We evaluate DrLLM in zero-shot scenarios by comparing it with four state-of-the-art LLMs, including GPT-4o-mini [29], Llama3-70b [4], Deepseek-chat-v2 [5] and Qwen2-57b-a14b-instruct [30]. We directly call their official application programming interfaces (APIs) in non-streaming mode to enable rapid response and kept the parameters as official default.

C. Prompts and Evaluation Metrics

As mentioned in Section III-C, we use regular expressions to extract classification probabilities from the output and constrain them using CoD. However, during the evaluation process, we found that the LLMs occasionally outputs in an incorrect format. In order to verify the effectiveness of multiple modules of DrLLM, we carefully crafted four prompt templates $P_0 = BP(BasicPrompt)$, $P_1 = BP + CoD$, $P_2 = BP + CoD + CoT$, $P_3^{'} = KP + BP + CoD$, $P_3 = KP + BP + CoD + CoT$ for comparison. Due to space constraints, we will present the full prompt templates in the open source repository.

In addition, in order to evaluate the performance of DrLLM on multiple models and multiple prompts, in this paper, we adopted F1, Recall and AUC as the evaluation metrics during our experiments.

D. Ablation Study

In order to analyze the effectiveness of DrLLM, we conduct a series of ablation experiments with other state-of-the-art LLMs on our proposed modules. As shown is Table I.

TABLE I
ACCURACY OF DRLLM WITH DIFFERENT LLM BACKBONES

Backbones	Metrics[1]	P_0	P_1	P_2	P_3'	P_3
GPT-4o-mini	F1	0.6573▼14.99%	0.7024▼9.16%	0.7197▼6.92%	0.7510▼2.87%	0.7732
	Recall	0.9033▲19.31%	0.6219▼17.86%	0.6620▼12.56%	0.7308▼3.47%	0.7571
	AUC	0.7208▼13.62%	0.7797▼6.57%	0.7957▼4.65%	0.8254▼1.09%	0.8345
Llama3-70b	F1	0.5117▼13.33%	0.5275▼10.65%	0.2951▼50.02%	0.5403▼8.49%	0.5904
	Recall	0.3567▼19.21%	0.3744▼15.20%	0.1737▼60.66%	0.5037▲14.09%	0.4415
	AUC	0.6613▼18.60%	0.8543▲5.15%	0.7614▼6.28%	0.6789▼16.43%	0.8124
Qwen2-57b-a14b-instruct	F1	0.6525▼12.25%	0.6396▼13.98%	0.6968▼6.29%	0.7090▼4.65%	0.7436
	Recall	0.8307▲9.95%	0.7448▼1.41%	0.7383▼2.27%	0.7554▼0.01%	0.7555
	AUC	0.6984▼13.69%	0.7038▼13.02%	0.7584▼6.27%	0.7310▼9.66%	0.8092
Deepseek-chat-v2	F1	0.7672▼9.73%	0.7904▼7.00%	0.8114▼4.53%	0.7759▼8.71%	0.8499
	Recall	0.6507▼20.77%	0.6994▼14.84%	0.6882▼16.21%	0.7256▼11.65%	0.8213
	AUC	0.8710▼5.45%	0.8853▼3.90%	0.8927▼3.09%	0.8787▼4.61%	0.9212

[1] We highlight the P_3 by ●.

[2] If other prompts' perform better than P_3 on a metric we denote it by ●, if not as well we denote it by ●.

Overall, Deepseek-chat-v2 performs the best and outperforms almost all other models under both P_0 and P_3, with F1 reaching 0.8499 and AUC 0.9212 under P_3, and also performs stably under P_1 and P_2, which we also use as the backbone of DrLLM. While the worst LLM is Llama3-70b, which is almost unusable in zero-shot scenario. In addition, all the models almost increase in each scenarios after adding our designed KP, CoD and CoT, and the classification ability is greatly enhanced in zero-shot scenarios, which also proves the effectiveness of Knowledge Embedding and Token Embedding in DrLLM from the side.

E. Stability Analysis

During the evaluation process, we checked the outputs of multiple models and found that the outputs of LLMs were not stable and occasionally exhibited unconstrained abnormal behaviors. We collected the abnormal outputs and defined the anomalies as follows:

1) Confidence Bias: We denotes this problem as L_1. In the classification, the model gives confidence for both abnormal and normal traffic, but the sum of their probability results is not 1. This type of problem is more frequent in the case of P_0, but after adding CoD, the probability of such problems is greatly reduced, which also indirectly proves the effectiveness of CoD in DrLLM in constraining the output of large models.

2) Confidence Lost: We denotes this problem as L_2. LLMs mean that analysis is impossible and there is a lack of confidence in the predicted probabilities based on the data.

Table II demonstrates the probability of occurrence of the L_1 and L_2 in each scenarios. In the P_0, the L_1 of GPT-4o-mini has the highest probability 0.8760, in contrast to DeepSeek-chat-v2 which has the best stability of the L_1 at only 0.0309. Overall, comparing to other LLMs, Deepseek-chat-v2 performs most stably in multiple scenarios, but after adding of multiple modules, the L_1 of GPT-4o-mini drops rapidly to 0.0014 and the L_2 to 0.0153, which is comparable to the performance of Deepseek-chat-v2, which is a side-effect of proving the DrLLM multiple modules' effectiveness.

TABLE II
ABNORMAL OUTPUT OF BACKBONES.

Backbones	Metrics[5,6]	P_0	P_1	P_2	P_3'	P_3
GPT[1]	L_1	87.60	18.07	0.37	0.33	0.14
	L_2	3.40	3.02	1.92	1.84	1.53
Llama3[2]	L_1	28.29	0	0.03	0.07	0.10
	L_2	0	0.48	0	0.17	0.10
Qwen2[3]	L_1	53.46	1.29	0.65	0.30	0.27
	L_2	14.03	9.31	2.45	0.97	0.92
Deepseek[4]	L_1	3.09	0	0	1.27	0.20
	L_2	7.07	3.09	1.46	1.19	0.90

[1] Denotes GPT-4o-mini. [2] Denotes Llama3-70b.
[3] Denotes Deepseek-chat-v2. [4] Denotes Qwen2-57b-a14b-instruct.
[5] We highlight the best value by ●.
[6] The unit of L_1 and L_2 is percentage [%]. Lower is better.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a DDoS resistance method based with Large Language Models which is called DrLLM. We embed the global information and local information with Knowledge Embedding and Token Embedding modules. We design the Progressive Role Reasoning for classification and explore the application of DrLLM in zero-shot scenarios with extensive experiments. The ablation experiments have demonstrated the effectiveness of the individual modules in DrLLM and proved the universality of LLMs in the networking domain. In the future, we will explore the application of retrieval-augmented generation (RAG) in LLMs.

REFERENCES

[1] M. A. Ferrag, F. Alwahedi, A. Battah, B. Cherif, A. Mechri, and N. Tihanyi, "Generative ai and large language models for cyber security: All insights you need," *arXiv preprint arXiv:2405.12750*, 2024.

[2] Imperva. (2024) 2024 imperva ddos threat landscape report. [Online]. Available: https://www.imperva.com/resources/resource-library/reports/2024-imperva-ddos-threat-landscape-report/

[3] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[4] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[5] X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu *et al.*, "Deepseek llm: Scaling open-source language models with longtermism," *arXiv preprint arXiv:2401.02954*, 2024.

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

[7] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[8] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[9] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, "A survey on multimodal large language models," *arXiv preprint arXiv:2306.13549*, 2023.

[10] S. Wang, L. Zhang, L. Zhu, T. Qin, K.-H. Yap, X. Zhang, and J. Liu, "Cog-dqa: Chain-of-guiding learning with large language models for diagram question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 13 969–13 979.

[11] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 international carnahan conference on security technology (ICCST)*. IEEE, 2019, pp. 1–8.

[12] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.

[13] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del Rincon, and D. Siracusa, "Lucid: A practical, lightweight deep learning solution for ddos attack detection," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 876–889, 2020.

[14] F. Ullah, S. Ullah, G. Srivastava, and J. C.-W. Lin, "Ids-int: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic," *Digital Communications and Networks*, vol. 10, no. 1, pp. 190–204, 2024.

[15] D. Barradas, N. Santos, L. Rodrigues, S. Signorello, F. M. Ramos, and A. Madeira, "Flowlens: Enabling efficient flow classification for ml-based network security applications." in *NDSS*, 2021.

[16] Z. Liu, H. Namkung, G. Nikolaidis, J. Lee, C. Kim, X. Jin, V. Braverman, M. Yu, and V. Sekar, "Jaqen: A high-performance switch-native approach for detecting and mitigating volumetric ddos attacks with programmable switches," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3829–3846.

[17] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time series forecasting by reprogramming large language models," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=Unb5CVPtae

[18] N. Gruver, M. A. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=md68e8iZK1

[19] R. Manvi, S. Khanna, G. Mai, M. Burke, D. B. Lobell, and S. Ermon, "GeoLLM: Extracting geospatial knowledge from large language models," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=TqL2xBwXP3

[20] L. Wu, Z. Qiu, Z. Zheng, H. Zhu, and E. Chen, "Exploring large language model for graph data understanding in online job recommenda-tions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 8, 2024, pp. 9178–9186.

[21] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[22] J. Liu, A. Liu, X. Lu, S. Welleck, P. West, R. L. Bras, Y. Choi, and H. Hajishirzi, "Generated knowledge prompting for commonsense reasoning," *arXiv preprint arXiv:2110.08387*, 2021.

[23] M. A. Ferrag, A. Battah, N. Tihanyi, M. Debbah, T. Lestable, and L. C. Cordeiro, "Securefalcon: Are we there yet in automated software vulnerability detection with llms?" *arXiv preprint arXiv:2307.06616*, 2024.

[24] Q. Li, Y. Zhang, Z. Jia, Y. Hu, L. Zhang, J. Zhang, Y. Xu, Y. Cui, Z. Guo, and X. Zhang, "Dollm: How large language models understanding network flow data to detect carpet bombing ddos," *arXiv preprint arXiv:2405.07638*, 2024.

[25] T. Wang, X. Xie, L. Zhang, C. Wang, L. Zhang, and Y. Cui, "Shieldgpt: An llm-based framework for ddos mitigation," in *Proceedings of the 8th Asia-Pacific Workshop on Networking*, 2024, pp. 108–114.

[26] J. Zhang, H. Bu, H. Wen, Y. Chen, L. Li, and H. Zhu, "When llms meet cybersecurity: A systematic literature review," *arXiv preprint arXiv:2405.03644*, 2024.

[27] M. X. Liu, F. Liu, A. J. Fiannaca, T. Koo, L. Dixon, M. Terry, and C. J. Cai, ""we need structured output": Towards user-centered constraints on large language model output," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–9.

[28] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[29] OpenAI. (2024) Gpt-4o mini: advancing cost-efficient intelligence. [Online]. Available: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[30] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang *et al.*, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024.