Exploring the Role of Large Language Models in Cybersecurity: A Systematic Survey

Shuang Tian, Tao Zhang, *Member, IEEE,* Jiqiang Liu, *Senior Member, IEEE,* Jiacheng Wang, Xuangou Wu, Xiaoqiang Zhu, *Member, IEEE,* Ruichen Zhang, *Member, IEEE,* Weiting Zhang, *Member, IEEE,* Zhenhui Yuan, *Senior Member, IEEE,* Shiwen Mao, *Fellow, IEEE,* Dong In Kim, *Life Fellow, IEEE*

Abstract—With the rapid development of technology and the acceleration of digitalisation, the frequency and complexity of cyber security threats are increasing. Traditional cybersecurity approaches, often based on static rules and predefined scenarios, are struggling to adapt to the rapidly evolving nature of modern cyberattacks. There is an urgent need for more adaptive and intelligent defence strategies. The emergence of Large Language Model (LLM) provides an innovative solution to cope with the increasingly severe cyber threats, and its potential in analysing complex attack patterns, predicting threats and assisting real-time response has attracted a lot of attention in the field of cybersecurity, and exploring how to effectively use LLM to defend against cyberattacks has become a hot topic in the current research field. This survey examines the applications of LLM from the perspective of the cyber attack lifecycle, focusing on the three phases of defense reconnaissance, foothold establishment, and lateral movement, and it analyzes the potential of LLMs in Cyber Threat Intelligence (CTI) tasks. Meanwhile, we investigate how LLM-based security solutions are deployed and applied in different network scenarios. It also summarizes the internal and external risk issues faced by LLM during its application. Finally, this survey also points out the facing risk issues and possible future research directions in this domain.

Index Terms—Large Language Model, Cybersecurity, Cyber attacks, Cyber defense, Intrusion detection, Anomaly detection, Phishing attack detection, Malware detection, Vulnerability detection, Vulnerability patch, Cyber Threat Intelligence, next-generation networks

I. INTRODUCTION

AS the information age develops rapidly, cyberattacks are taking on the characters of high frequency, diversity and complexity [1]–[4]. Critical infrastructure and personal sensitive data are facing a broad range of novel threats, including malware, ransomware, and DDoS attacks [5]–[7]. The evolution of threat methods and the growing intensity of these threats is resulting in severe economic and property damage [8]. In the United States, a ransomware attack on Colonial Pipeline system completely shuttered their operations, leading to a gasoline shortage across the country's East Coast for an entire week. In 2022, Sunwing Airlines were forced to cancel 188 flights and leave passengers stranded at airports for over three days, all due to a cyber attack on their supplier's systems. These are just a few of an ever-growing list of cyberattacks that are altering different spheres of daily life, making cybersecurity one of the core foundational issues of modern global security concern [9].

As cyberattacks continue to evolve in their persistence, stealth, and unpredictability, existing cybersecurity measures struggle to keep pace in detecting, preventing and mitigating threats to networks. Traditional network defence methods based on fixed rules and scenarios have been exhausted in the complex network environment [10]. Although advanced AI-based cyber defence methods have developed rapidly in recent years with the rise of neural networks and deep learning technologies. However, these methods suffer from high false positives and lack of interpretability when put into commercial use [11], making it still a huge challenge to deal with rapidly evolving forms of cyber-attacks. Additionally, cybersecurity researchers are confronted with numerous challenges due to the performance limitations of these systems, including managing large volumes of sensitive data and dealing with the complexities of cybersecurity tasks [3]. In response to this situation, cybersecurity researchers recognise the need for stronger, more adaptable and smarter solutions to deal with the ever-increasing threat of attacks.

S. Tian is with the School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China. E-mail: tianshuang@bjtu.edu.cn.

T. Zhang, J. Liu and X. Zhu are with the School of Cyberspace Science and Technology, Beijing Jiaotong University, Beijing 100044, China. E-mail: taozh@bjtu.edu.cn; jqliu@bjtu.edu.cn; xqzhu@bjtu.edu.cn.

J. Wang is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798. E-mail: jiacheng.wang@ntu.edu.sg.

X. Wu is with the School of Computer Science and Technology, Anhui Province Key Laboratory of Digital Twin Technology in Metallurgical Industry, Anhui University of Technology, Anhui 243002, China. E-mail: wuxgou@ahut.edu.cn.

R. Zhang is with the College of Computing and Data Science, Nanyang Technological University, Singapore. E-mail: ruichen.zhang@ntu.edu.sg.

W. Zhang is with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China. E-mail: wtzhang@bjtu.edu.cn.

Z. Yuan is with the School of Engineering, University of Warwick, CV4 7AL Coventry, U.K. E-mail: zhenhui.yuan@warwick.ac.uk.

S. Mao is Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA. E-mail: smao@auburn.edu.

D. I. Kim is with the Department of Electrical and Computer Engineering, College of Information and Communication Engineering, Sungkyunkwan University, Suwon 16419, South Korea. E-mail: dongin@skku.edu.

The emergence of LLM has provided a new way of thinking about network defence. In recent years, LLMs have achieved significant breakthroughs in the field of natural language processing and have also shown great potential in cybersecurity defense. By leveraging extensive training datasets, LLMs can identify latent attack patterns and vulnerabilities, assist in analyzing attack behaviors, predict threats, and even provide real-time defensive support. They are capable of identifying cybersecurity risks based on historical attack data or contextual attack information and can proactively generate response strategies. LLM is used for a variety of cybersecurity tasks, such as threat detection, analysis of cybersecurity reports and the provision of defence recommendations.

In view of the substantial value of LLMs currently show in cybersecurity, we would like to provide an overview of present-day applications of LLMs in this domain, in order to provide future researchers with an outlook and ideas. In summary, the contributions made in this article are as follows:

- The feasibility and application prospects of LLMs in cybersecurity are discussed in depth through a systematic investigation of current benchmarking studies that assess the performance of LLMs and specific technical means to optimise the behaviour of LLMs in cybersecurity tasks.
- An innovative and comprehensive analysis of the defensive role played by LLM from the attacker's point of view across the various lifecycles of a cyberattack. Additionally, due to the important intelligence base role played by CTI in defence operations, and as a complement, we have also explored the defence role played by LLMs in CTI work.
- We also analyze the deployment and application approaches of LLM-based security applications in different network scenarios and the challenges they face
- This paper analyses the external and internal risks that LLMs may face in the process of executing cybersecurity tasks and provides risk warning and coping ideas for related research and applications.

The organizational structure of this article is illustrated in Fig. 1. Section II introduces the foundations of the LLM and provides a concise survey of existing investigations into the application of LLMs for cybersecurity, while also identifying and analyzing current research gaps. Section III briefly reviews research evaluating the performance of LLM in cybersecurity and optimising LLM technology in this field. Section IV introduces the network attack model employed. Sections V explores the applications and limitations of LLMs in different phases of a network attack lifecycle. Section VI explores the applications of LLMs in CTI. Section VII explores the deployment and application of LLM-based network security solutions in different network scenarios. Section VIII summarizes the internal and external risks associated with applying LLMs in the network security domain. Section IX summarises the current challenges encountered by LLMs in cybersecurity tasks and future research directions in this domain. Section X summarises our research. Table I lists and describes the acronyms used throughout this paper.

TABLE I
LIST OF ACRONYMS USED THROUGHOUT THIS PAPER.

Acronym	Definition
LLM	Large Language Model
CTI	Cyber Threat Intelligence
IDS	Intrusion Detection Systems
NER	Named Entity Recognition
ICS	Industrial Control Systems
PEFT	Parameter-Efficient Fine-Tuning
EDR	Endpoint Detection and Response
MHA	Multi-Head Attention
CoT	Chain-of-Thought
PbE	Programming by Example
EMAD	Evidence-Based Multi-Agent Debate
GNN	Graph Neural Network
RAG	Retrieval-Augmented Generation
CEC	Contract-External Function-Call
IoT	Internet of Things
USE	Unidirectional Semantic Extractor
BSE	Bidirectional Semantic Extractor
ML	machine learning
BS	base station
NF	network function
LEGD	Large Language Model-Enhanced Graph Diffusion
DTN	digital twins network
SAGIN	satellite-aerial-ground integrated network
SLM	Small Language Model

II. BACKGROUND AND RELATED WORK

In this section, we will briefly introduce the knowledge about LLM that is required to read this paper, and then review the relevant review literature on the current research on the application of LLM in cybersecurity in order to present the Research gap that currently exists.

A. Foundations of LLMs

As a cutting-edge technology in the field of artificial intelligence, LLM has been widely used in many fields and has become one of the current research hotspots. Its core technology is built on the ransformer architecture [3], which can be divided into three typical structures based on decoding strategies [12]:

- The encoder-only architecture is excellent at language comprehension tasks, and is usually used for linguistic feature extraction, and the BERT model is a representative model of this architecture.
- The encoder-Decoder architecture is widely used in sequence-to-sequence tasks, and is widely used in text translation and speech recognition.
- The decoder-only architecture is the current research hot architecture, and the popular GPT series is based on this architecture.

After pre-training with large-scale datasets, LLMs can acquire language comprehension and logical reasoning abilities, and if trained with domain-specific datasets during the pre-training process, LLMs can even perform comparably to humans in some domains. In addition, fine-tuning and prompt engineering techniques are important complementary means to optimise
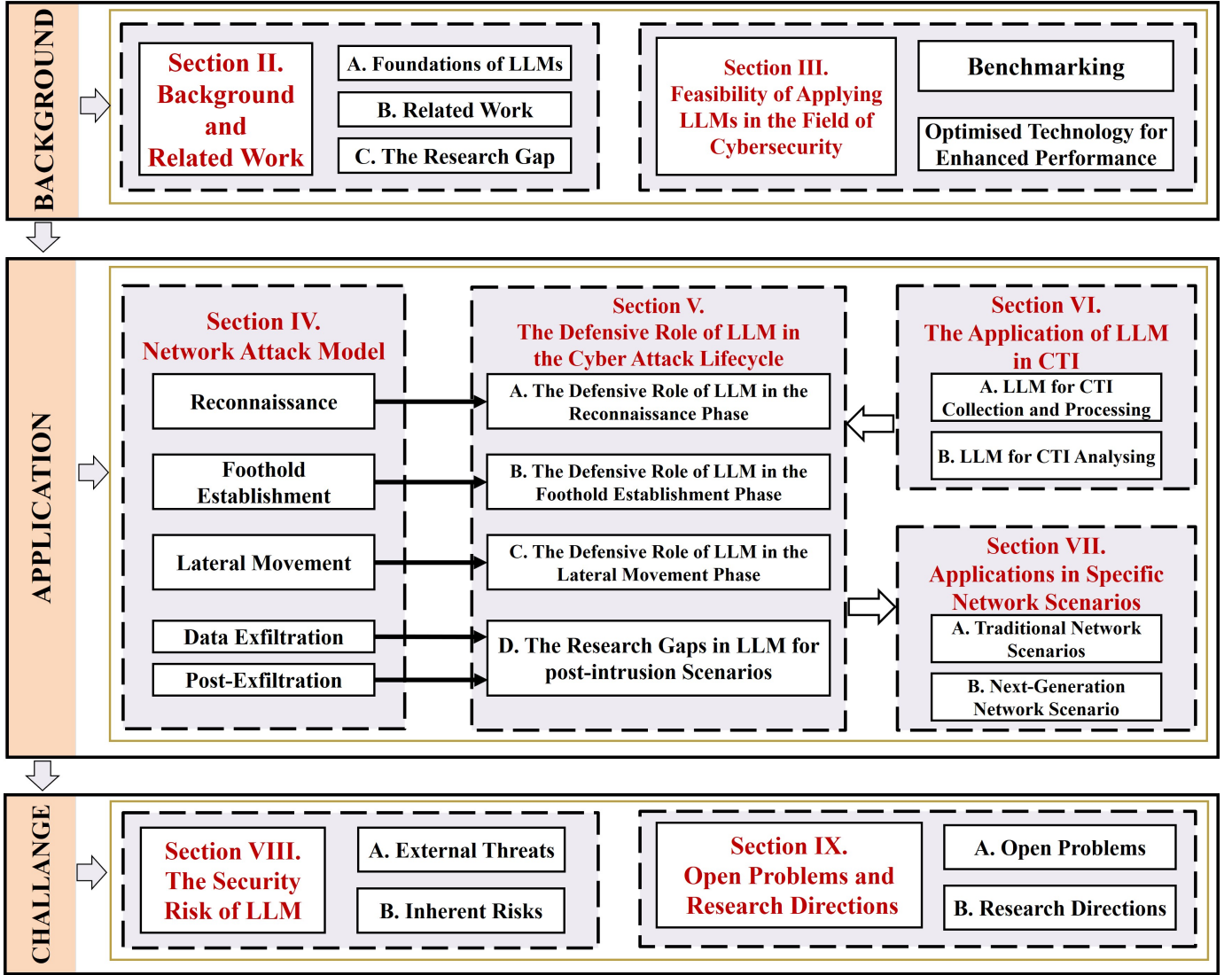
Fig. 1. The overall organizational structure of this survey.

the model's ability to perform domain-specific tasks in the post-pretraining phase, and the use of these two techniques can effectively expand the scope of application of pre-trained models. However, it should be noted that there are still some challenges that hinder the application scope and performance of LLM, such as high quality domain training sets, high training costs and inference delay issues.

B. Related Work

Current review studies on LLM in network security usually focus on the application methods, usage scenarios, and performance evaluation of LLM. In this subsection, we review these studies and describe the contributions and insights they provide.

1) Focused Analysis of LLMs in Specific Cybersecurity Tasks: Some surveys have focused its research on the performance of LLMs in specific cybersecurity tasks in order to deeply analyse their capabilities and limitations in practical applications. For instance, Ref. [12] systematically combs

through the research on LLM-based intrusion detection systems (IDSs) in different architectures and deployment environments while demonstrating their practical utility through real-world use cases. However, the article also points out that LLM-based IDS systems still face many challenges, such as training data privacy issues, network data heterogeneity, and inherent security vulnerabilities in LLM architectures. Ref. [13] investigates the current state of the application of LLMs in vulnerability detection. The authors point out that the dominant model architecture in the field is shifting from encoder-only to decoder-only. In addition, existing research is overly reliant on C/C++ language vulnerability datasets and lacks repository-level data, which limits LLM's ability to generalise across languages and detect complex multi-file vulnerabilities. Ref. [14] summarises the main types of LLMs and the main LLM performance optimisation techniques in vulnerability detection and repair. The article also points out that there is a lack of class-level or repository-level training datasets in this area, the reliability of existing datasets is poor, and the datasets

TABLE II
COMPARISON OF LLM APPLICATION SURVEYS IN CYBERSECURITY. "●" AND "○" REPRESENT RXPLORED AND NOT RXPLORED, RESPECTIVELY.

Surveys	Defense against Reconnaissance Attack	Defence against Foothold Establishment Attack	Defence against Lateral Movement Attack	Cyber Threat Intelligence Work	Application in Different Network Scenarios	LLM's Own Security Risks
Kheddar [12]	○	○	●	○	●	●
Sheng *et al.* [13]	○	●	○	○	○	○
Zhou *et al.* [14]	○	●	○	○	●	○
Zhang *et al.* [15]	○	●	●	●	○	●
Hang *et al.* [16]	○	●	●	●	○	○
Motlagh *et al.* [17]	●	●	●	○	○	○
Chen *et al.* [3]	●	●	●	●	○	○
Yao *et al.* [18]	○	●	○	○	○	●
Our survey	●	●	●	●	●	●

often lack test samples. Finally, the authors argue that much of the current research does not emphasise integration with developer workflows, and that there is a lack of mechanisms for interaction between users and LLMs.

2) Broad Exploration of LLMs in Multi-Domain Cybersecurity Applications: Some surveys have taken a broader perspective to deeply analyze the performance of LLMs across multiple key tasks for cybersecurity. Both Ref. [15] and Ref. [16] provide systematic summaries and organization of current research on the application of LLMs in cybersecurity. While the two surveys unanimously acknowledge that LLMs can significantly improve the efficiency of cybersecurity tasks, they also highlight persistent challenges, including external attack threats and inherent limitations in model performance. Within the the National Institute of Standards and Technology Cybersecurity Framework, Ref. [17] studies LLM applications in the identify, protect, detect, respond, and recover phases. It notes that current research focuses on protect and detect scenarios, but post-attack scenarios, including response and recovery phases, remain understudied. Given their critical roles, expanding LLM research in these areas is essential for comprehensive cybersecurity. Ref. [3] examines LLM applications in four key threat detection areas: CTI, textual threat detection, malware detection, and intrusion discovery. It reveals that LLMs surpass traditional methods primarily in specific tasks like NER, Relation Extraction, and structured information processing. However, for significantly more sophisticated threat detection scenarios, LLMs typically necessitate integration with complementary technologies for optimal performance. Ref. [18] examines the application of LLMs in code security tasks, such as secure coding and vulnerability detection. The survey reveals that LLM-based approaches generally surpass traditional methods in this domain, although they exhibit higher rates of both false negatives and false positives. Furthermore, through an investigation of LLMs' application in data security tasks—encompassing data integrity, confidentiality, and reliability—the research demonstrates that LLMs not only minimize manual intervention but also achieve superior performance in these areas. Both Ref. [15] and Ref. [16] provide systematic summaries and organization of current research on the application of LLMs in cybersecurity. While

the two surveys unanimously acknowledge that LLMs can significantly improve the efficiency of cybersecurity tasks, they also highlight persistent challenges, including external attack threats and inherent limitations in model performance. Within the the National Institute of Standards and Technology Cybersecurity Framework, Ref. [17] studies LLM applications across identify, protect, detect, and respond stages. While current research predominantly focuses on Protect and Detect, post-attack scenarios, encompassing the Respond and Recovery phases, remain significantly understudied. Given their important role, expanding LLM research in these areas is essential for comprehensive cybersecurity. Ref. [3] examines LLM applications in four key threat detection areas: Cyber Threat Intelligence (CTI), textual threat detection, malware detection, and intrusion discovery. It reveals that LLMs surpass traditional methods primarily in specific tasks like NER, Relation Extraction, and structured information processing. However, for significantly more sophisticated threat detection scenarios, LLMs typically necessitate integration with complementary technologies for optimal performance. Ref. [18] examines the application of LLMs in code security tasks, such as secure coding and vulnerability detection. The survey reveals that LLM-based approaches generally surpass traditional methods in this domain, although they exhibit higher rates of both false negatives and false positives. Furthermore, through an investigation of LLMs' application in data security tasks—encompassing data integrity, confidentiality, and reliability—the research demonstrates that LLMs not only minimize manual intervention but also achieve superior performance in these areas.

C. The Research Gap

The research of the existing review papers in this section focuses on evaluating the performance of LLMs in specific tasks or application scenarios. However, for the systematic defence role of LLM in the whole network attack and defence process, there is still a certain research gap in current research. This gap has resulted in an incomplete understanding of the overall efficacy of LLMs in more holistic and dynamic cyber defense contexts. In addition, as a key intelligence component

in cybersecurity defence, current research on CTI usually evaluates the performance of LLMs in isolated CTI tasks and lacks the defence role that LLMs can play from the whole CTI lifecycle. On the other hand, most studies nowadays have also overlooked the deployment of LLM in applications, and the lack of research here may create obstacles for future real-world applications. The surveys covered in this section are shown in Table II.

Consequently, this survey seeks to bridge this gap by adopting an innovative perspective rooted in the attacker lifecycle. It systematically and comprehensively examines the defensive role of LLMs at various stages of the cyber attack lifecycle. Additionally, it evaluates their performance in different stages of CTI tasks for effective integration with real-world defense actions. Also investigating the way in which LLM-based security solutions are deployed at the time of application. This comprehensive analysis not only enhances the understanding of the practical value of LLMs in real-world cyber defense environments but also offers solid theoretical support for further research and the execution of defense measures.

III. FEASIBILITY OF APPLYING LLMS IN THE FIELD OF CYBERSECURITY

Although LLMs are widely used in many fields [19]–[21], due to the highly professional and complex of cybersecurity tasks, it is still questionable whether LLMs can perform these tasks efficiently. To explore this issue, researchers evaluated the performance of LLMs in cybersecurity tasks through benchmarking and further explored techniques to optimise the performance of LLMs in cybersecurity tasks.

Liu [22] introduced SecQA, a benchmarking tool designed to evaluate LLM performance in computer security. Liu used GPT-4 to generate two multiple-choice question sets, v1 and v2, based on the content of a computer security book. v1 focused on LLM's basic understanding and application of cybersecurity knowledge, while v2 examined LLM's more in-depth and comprehensive understanding of advanced security topics through the use of more complex and detailed questions. Through experimental evaluation, the results show that GPT-3.5-Turbo and GPT-4 maintain high accuracy rates on the v2 set. Tihanyi *et al.* [23] constructed the CyberMetric-80 dataset to evaluate LLMs' cybersecurity knowledge coverage, which underwent rigorous expert validation to ensure answer accuracy. In a controlled evaluation, multiple LLMs and human participants completed the CyberMetric-80 assessment. The findings indicated that LLMs, especially GPT-4o and GPT-4-turbo, exhibited expertise comparable to seasoned cybersecurity professionals. The results of these two benchmark tests suggest that LLMs have a strong foundation in cybersecurity knowledge and can effectively comprehend and apply it.

Liu *et al.* [24] evaluated generative LLMs in cybersecurity using a multi-task framework benchmarking with 10 datasets corresponding to four representative security tasks: NER, summarization, multiple choice, and text classification. Bhusal *et al.* [25] proposed the SECURE benchmarking framework that can be used to assess the capabilities of LLMs in industrial control systems (ICS) security consulting within three key abilities:

- **Extraction**: Evaluates the efficiency of information retrieval using datasets from MITRE ATT&CK and CWE.
- **Understanding**: Employs the Vulnerability Out-Of-Distribution test set to determine whether models can identify unanswerable questions in the absence of contextual information.
- **Reasoning**: Uses the Risk Evaluation Reasoning Task, constructed from CISA ICS security reports, to assess models' reasoning abilities in risk evaluation.

Using the two previously mentioned benchmark frameworks, multiple LLMs were assessed, with models like GPT-4 scoring highly, indicating that LLMs still perform well on specific security tasks.

Although several studies have demonstrated the potential of LLMs for cybersecurity applications through benchmarking, most LLMs are not specifically designed for this domain task and may suffer from performance degradation due to lack of domain knowledge.This issue can be addressed with specialized techniques that can significantly improve their performance. Fine-tuning, prompt engineering, and domain-specific pre-training have been demonstrated to improve the performance of LLMs in this domain [15]. For example, Zhang *et al.* [26] developed tailored instructions and conversations for cybersecurity fine-tuning and applied LoRA fine-tuning to baseline LLMs, achieving a 10%–25% performance improvement. Similarly, Siracusano *et al.* [27] utilised specially designed prompts to guide their structured CTI extraction framework, aCTIon, thereby reducing the hallucinations when dealing with complex CTI data. In another study, Liu *et al.* [24] introduced CyberDirective, a generative LLM that is fine-tuned on the CyberBench dataset using instruction tuning and parameter-efficient fine-tuning (PEFT), and demonstrated excellent in multiple cybersecurity tasks with excellent performance. The studies mentioned above highlight the efficacy of the specialized techniques in enhancing LLM's domain-specific capabilities, particularly in cybersecurity applications.

Although the specialized and complex character of cybersecurity tasks challenges the applicability of LLM, many studies have shown that LLM already exists in a wide range of applications in several cybersecurity sub-domains and exhibits great potential for application. Meanwhile, the development of techniques such as fine-tuning, domain-specific pre-training, and prompt ngineering provides strong support for improving the performance of LLM in cybersecurity tasks. Results from various benchmark and performance tests also indicate that LLMs are increasingly capable in terms of understanding, reasoning, and knowledge coverage, gradually meeting the practical demands of cybersecurity work. Thus, it can be stated that the implementation of LLMs in cybersecurity is progressing at a high pace and will be key role in improving productivity, automating analysis, and facilitating decision making in the future.

IV. NETWORK ATTACK MODEL

In order to easily describe the role played by LLM in each phase of cyber defence, it is necessary first to build a cyber attack model to describe the cyber attack process.
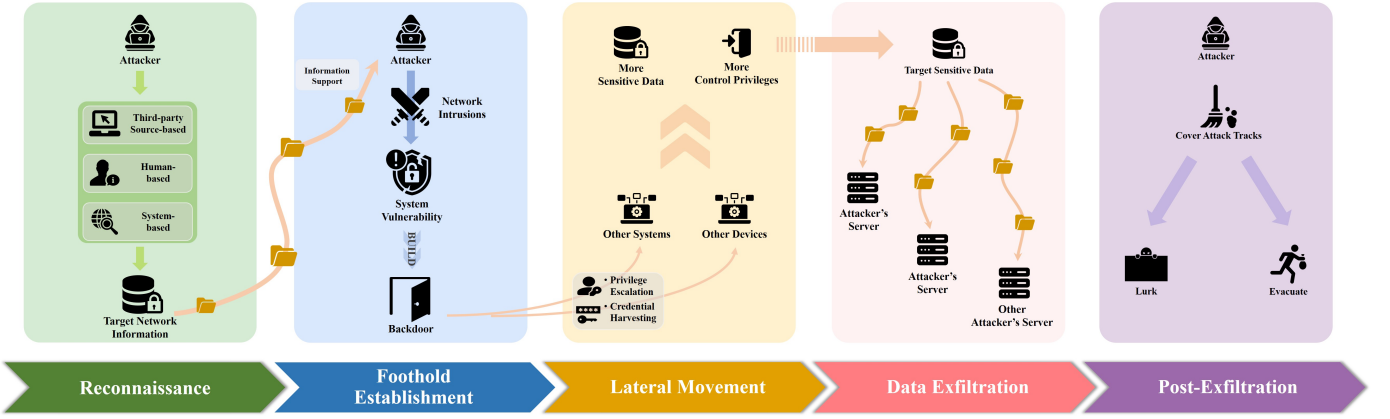
Fig. 2. Network attack model.

In this paper, we divide the life cycle of an external cyber attack into five phases: reconnaissance, foothold establishment, lateral movement, data exfiltration, and post-exfiltration [28], as shown in Fig. 2.

Reconnaissance refers to the process by which an attacker gathers information about the target network at the beginning of an attack. During this stage, attackers usually use covert passive reconnaissance methods [28]. At this stage, attackers may collect information from public resources on the Internet, such as WHOIS websites [29] and the Google Hacking Database [30]. Attackers may also use social engineering techniques to obtain information from users of the target network [31], such as phishing attacks, enticing users of the target network to insert physical media with viruses into their computers, causing their computers to be infected with viruses [31], and directly physically intruding such as tailgating attacks. In addition, attackers may also use technical methods to interact directly or indirectly with the target computer system to collect information, such as TCP scanning, ARP scanning, UDP scanning [32] to obtain information about the target system.

The foothold establishment phase signifies that the attacker has successfully infiltrated the target network. Attackers often create a "foothold" within the target network to maintain long-term access to the target network system. Standard techniques include exploiting known vulnerabilities or zero-day vulnerabilities in web applications [33], using spear-phishing attacks to implant a backdoor on the target's endpoint device [34], or carrying out watering-hole attacks [35], where attackers infect websites frequently visited by the target network's users, injecting malicious code into the victim's devices.

The lateral movement phase aims to expand the attacker's access to the target network to support the attacker in obtaining more sensitive data and control privileges of the target system. In this phase, the attacker usually uses Privilege Escalation, Credential Harvesting, and other means to expand the scope of his activities in the target network. In this phase, the techniques usually used are using the vulnerability of the target network system to enhance the attacker's privileges in the target system, using credentials dumping [36], hash transfer attacks [37] and session stealing techniques [38] to illegally obtain the user credentials of the target network to obtain a broader range

of access privileges. Moreover, it has become challenging to completely eliminate the attacker from the system at this phase because the attacker has deep roots in the network and gained persistent control.

In the data exfiltration phase, the attacker transmits the stolen data to an external server under his control. In many network systems, firewalls and other security measures often focus on filtering incoming traffic, with little or no outbound traffic monitoring. This design facilitates data transfer, allowing attackers to easily pass sensitive information back. Typically, attackers will divide the collected data into multiple batches and send them to different servers in their organisation, which greatly reduces the possibility of the transmission being discovered by the defender.

The post-exfiltration phase is the action taken by the attacker after completing the attack's intended target. Typically, the attacker has two choices: the first is to evacuate the target network system and cover up the traces of the attack as much as possible, deleting any evidence left in the system; the second is to choose to continue to lurk in the target system in order to carry out new attacks in the future. This type of attacker usually hides their control privileges to maintain access to the target network for a long time without being detected.

There is a near linear flow relationship between the above five phases, but sometimes the attacker may also continue the previous phase of the task in parallel when proceeding to the next phase for reasons such as lack of preparation for the attack. For instance, the attacker may continue to carry out reconnaissance work while establishing a foothold.

V. THE DEFENSIVE ROLE OF LLM IN THE CYBER ATTACK LIFECYCLE

In this section, we will talk in detail about how LLM plays a defensive role in the various lifecycles of a network attack.

A. The Defensive Role of LLM in the Reconnaissance Phase

Reconnaissance attacks are the beginning stage of cyberattacks [39], in which the attacker's reconnaissance behaviour is usually highly dispersed and covert, which poses a challenge

TABLE III
THEMATIC WORKS ABOUT LLMs ON DEFENSE AGAINST RECONNAISSANCE ATTACKS.

Task	Detection Target	Detection Approach	Techniques	Literature
System-based Reconnaissance Detection	Detect the advanced adversaries in **Smart Satellite Networks**.	**Transform network data** into contextually suitable inputs to **capture contexts and long-range relationships**.	Transformer-based MHA	[40]
	Detect the **malicious user activities**.	Use **three self-designed agents** to detect attack behaviors by **detecting log files**.	CoT reasoning, PbE paradigm, EMAD mechanism	[41]
	Attack **early detection**, threat intelligence gatherin, and analysis of attacker's behavior.	Use **LLM-based honeypot** to simulate realistic shell responses and manages sessions for every attacker.	CoT prompting, Few-shot learning, Session management.	[42]
Human-based Reconnaissance Detection	Detect malicious web pages.	Use the question-and-answer **detection example** to guide the LLM.	K-means clustering, Few-shot Prompting	[43]
	Detect **phishing email**.	Translate email into **LLM-readable** format and use **CoT** to guide the LLM.	Prompt Engineering, CoT prompting	[44]
	Detect **malicious behaviours** in the code.	Use prompt and Provide **complete contextual information**.	Prompt Engineering	[45]
	Detect and classify **malware**.	Use **few-shot** and **episodic training** to enhance LLM malware detection and classification.	Few-shot learning, Episodic training	[46]

to the defender's detection and blocking efforts. The reconnaissance attacks in this phase can be classified into three types: third-party source-based reconnaissance, which obtains attack information from third parties (e.g. third-party websites and dark web), human-based reconnaissance, which obtains attack information from the target network users, and system-based reconnaissance, which obtains attack information from the target computer system (hardware or software) [32]. LLM can effectively detect human-based and system-based reconnaissance attacks, helping network defenders better prevent the leakage of their sensitive information. All related works in this subsection are summarized in Table III.

1) LLM for System-based Reconnaissance Attack Detection: In reconnaissance attacks targeting information systems, attackers frequently utilize remote scanning or sniffing to extract sensitive data from targeted systems. While these reconnaissance activities inevitably generate detectable traces within the target system, conventional detection approaches relying on rule-based or signature-based methodologies demonstrate limited effectiveness against sophisticated attack patterns. Conversely, LLMs, leveraging their advanced pattern recognition capabilities, demonstrate substantial advantages in both the precise identification and predictive analysis of reconnaissance attack behaviors.

Hassanin *et al.* [40] proposed a pre-trained LLM, PLLM-CS. The model initially generated sentences from the multivariate token series in the network traffic data. It then divided these sentences into tokens to capture contexts and long-range relationships within the traffic through the Transformer-based Multi-Head Attention (MHA) mechanism, which helped identify fragmented and frequent probing behaviors during attack detection. Song *et al.* [41] proposed Audit-LLM, a framework for detecting external attacks through log analysis, which consists of three agents: a Decomposer, a Tool Builder, and an Executor. The Decomposer uses Chain-of-Thought (CoT) reasoning to decompose complex tasks into subtasks.

The Tool Generator generates Python tools using programming by example (PbE) paradigms, ensuring reliability through testing and refinement. The executor completes subtasks using CoT reasoning and employs paired evidence-based multi-agent debate (EMAD) mechanisms to reduce LLM illusions, iteratively optimizing results until consensus is reached.

In addition, there is now some research into using LLMs to create new types of honeypot systems. LLMs can use their knowledge base and memory capabilities to create deceptive system environments for different attackers to slow down the detection of the attackers. Sladić *et al.* [42] proposed shelLM, a shell-based honeypot software using LLM. This honeypot uses the CoT prompting and few-shot learning and it can generates responses that are consistent with a real Linux shell based on the interaction history and the attacker's commands. Meanwhile, due to the non-deterministic nature of LLM, shelLM can simulate multi-user environments and enhance the realism of the honeypot system. Evaluations conducted by volunteers show a true negativity rate of 0.9, indicating that it can effectively mimic the responses of a real system and deceive users in 90% of the cases.

2) LLM for Human-based Reconnaissance Attack Detection: Social engineering is a widely used technique in cyberattacks, often resulting in significant data breaches [31]. Attackers typically employ methods such as phishing and watering-hole attacks to deceive users and extract sensitive system information. Recent advancements in LLM technology have shown promising results in detecting phishing attacks and identifying malware, offering innovative technical solutions to combat these social engineering threats.

Malicious web pages and phishing are two common means of detecting attacks. Li *et al.* [43] introduced Prompt-URL, a few-shot prompting approach for LLM-based malicious webpage detection. The method reformulates the detection task as a question-answering framework, where URL and website content serve as the question, and the webpage's malicious

classification as the answer. Using Sentence-BERT, the model extracts vector representations of questions, applies semantic clustering to refine question-answer pairs, and utilizes them as LLM prompts. Koide *et al.* [44] developed ChatSpamDetector, an LLM-based system for phishing email detection. The system preprocesses emails by reconstructing and simplifying their content for LLM analysis. Through prompt engineering, it assigns the LLM a spam detection role and employs CoT prompting to break the analysis into sub-tasks, guiding the LLM step-by-step through the analysis process while incorporating social engineering examples.

Malware is also a common way of reconnaissance attacks. Malware will obtain sensitive information by running malicious code on the user's system [47]. Using LLM can effectively identify malware and prevent the leakage of sensitive user information. Fang *et al.* [45] investigated LLM capabilities in defensive static analysis through a case study. Using prompt engineering, they guided GPT-4 to analyze both benign and malicious GitHub repositories, where it effectively identified malicious behaviors by recognizing characteristic patterns. Additionally, GPT-4 successfully detected malicious behaviors in the decompiled code of an Android msg-stealer virus within a comprehensive contextual framework. Stein *et al.* [46] proposed an LLM-based framework for malware detection and classification, utilizing a self-attentive mechanism to capture contextual patterns in packet sequences for distinguishing between benign and malicious traffic. Through few-shot learning, the framework effectively recognized novel malware with minimal labeled samples by generating class-specific prototypes and employing episodic training.

LLM shines in identifying hidden reconnaissance attacks due to its powerful data processing and pattern recognition capabilities. It also performs well in phishing defence due to its powerful knowledge base and logical reasoning capabilities. In addition, we also noticed that some studies have pointed out that LLM can also be used in network security education [47]–[49], which is also an application of LLM in defending against reconnaissance attacks.

B. The Defensive Role of LLM in the Foothold Establishment Phase

Vulnerability attacks are the most dominant methods of attack in the foothold phase, where an attacker will use specific vulnerabilities to successfully hack into a system and establish a foothold at the edge of the network to carry out subsequent attacks [50]. Timely patching vulnerabilities in the system to reduce the number of exploitable vulnerabilities can significantly reduce the probability of successful intrusion by attackers. With its robust knowledge base and analysis capabilities, the LLM can efficiently complete vulnerability detection, analysis and patching and other defensive work, thus significantly increasing the speed of vulnerability remediation and reducing the workload of network defenders. All related works in this subsection are summarized in Table IV.

1) LLM for Vulnerability Detection: High-speed software development technology has improved the efficiency of software development, but at the same time, it has also led to a significant increase in the number of vulnerabilities, which has brought new challenges for vulnerability detection. Recently, LLM has shown great potential in the field of vulnerability detection and has provided new ideas and methods to solve the vulnerability detection problem.

Lu *et al.* [51] proposed a new approach for vulnerability detection using LLM, GRACE. The proposed method helped LLM to capture more code structure information by generating a code property graph of the detected code, and identified the most relevant example code to the detected code from the codebase by comparing the semantic, lexical, and syntactic similarities to provide a better demonstration in the contextual learning of LLM. Yang *et al.* [52] developed MSIVD, a multitasking self-guided LLM model for vulnerability detection.They used PEFT and QLoRA techniques, fine-tuned by teacher-student dialogues, and integrated a graph neural network (GNN) to analyze the code data flow through control flow graphs. The GNN served as a lightweight adapter layer and concatenated learned embeddings at each training iteration with the hidden states of fine-tuned LLM along its last dimension. The last hidden states of an LLM encapsulated the information for all input elements before model prediction.

2) LLM for Vulnerability Analysis: LLM also demonstrates great potential in analysis work. Its ability to quickly identify and understand complex code structures provides defenders with timely and effective patch suggestions. This greatly shortens the cycle of vulnerability analysis and significantly improves the security and reliability of software.

Zhang *et al.* [53] introduced VTT-LLM, a framework for mapping vulnerabilities to tactics and techniques. To enhance the LLM's reasoning capability, the authors decomposed the mapping process into four sequential steps: vulnerabilities, weaknesses, attack patterns, and ATT&CK techniques, which are integrated as chained data during fine-tuning. Yin *et al.* [54] proposed a framework for predicting the exploitability of vulnerabilities based on vulnerability description information. This framework applies the BERT model through transfer learning, converting tokenized wordpiece lists into embedding vectors to capture the semantic information of the wordpieces and support subsequent predictive analysis. During the fine-tuning process, the model receives the segmented vulnerability description text as input and generates token embeddings layer by layer to capture multi-level semantic information. Luo *et al.* [55] introduced FELLMVP, a framework for categorizing smart contract vulnerabilities. They analyzed smart contract files to generate contract-external function-call (CEC) files that capture their semantic and structural content. These CEC files were then used to divide the dataset into eight subsets, each representing a different vulnerability type to ensure diversity. Small-batch incremental fine-tuning is performed using the LoRA method to obtain eight LLMs that specialize in identifying different vulnerability types.

In addition to the above mentioned analysis tasks, LLM has also been applied in many tasks such as penetration testing [60], vulnerability description generation [61], and vulnerability localization [62] and so on. We believe that LLM has a broad application prospect in vulnerability analysis, and more applications based on LLM will emerge in the future to

TABLE IV
THEMATIC WORKS ABOUT LLMs ON DEFENSE AGAINST FOOTHOLD ESTABLISHMENT ATTACKS.

Tasks	Approaches	Targets	Prompt Scenario	Literature
Vulnerability Detection	Code property graph, Contextual learning	Detecting software vulnerability	Identity, domain, in-context learning demonstrations and graph structure information	[51]
	Multitask self-instructed fine-tune, Situational dialogue	Detecting security vulnerability	Code snippets and basic task objectives	[52]
Vulnerability Analysis	Chain templates	Building VTT mapping	Analysis objectives and vulnerability description	[53]
	BERT migration learning, LSTM classification	Predicting exploitability of vulnerabilities	Analysis objectives description	[54]
	LLM-based parallel vulnerability analysis framework	Identifying vulnerabilities	Analysis objectives description	[55]
Vulnerability Patching	Prompt engineering, Static Analysis	Patching smart contract Vulnerability	Role-playing, task description, external structural information, Expected Output	[56]
	Leverage generative AI to create guiding prompts	Patching microarchitectural side-channel vulnerabilities	Identity and task information and prompts for the type of vulnerability	[57]
	Prompt engineering, Fine-tuning	Porting hard fork patches	patching objectives description	[58]
	In-context learning, Prompt engineering	Evaluating patches	Similar patches, bug descriptions, execution traces, failing test cases, test coverage and test patch	[59]

assist security personnel to complete the work of vulnerability analysis.

3) LLM for Vulnerability patch: Vulnerability patching, as a key aspect of network security, has long faced the challenges of efficiency bottlenecks and resource constraints. LLM, with its powerful code understanding and generation capabilities, provides new technical ideas for automated vulnerability repair work. Wang *et al.* [56] proposed a vulnerability remediation method based on the CoT mechanism, which guided LLMs to generate patches by decomposing the fixing task into a series of sub-tasks. The method also integrates static analysis techniques, including dependency analysis and program slicing, to assist LLMs in accurately locating vulnerabilities. Tol *et al.* [57] presented an automated framework for patching microarchitectural side-channel vulnerabilities using LLM. The framework integrated Microwalk [63] to locate the vulnerability and determine the cause of the vulnerability, and then utilized generative AI to craft prompts that guide LLM to deal with vulnerability. Through an iterative improvement process, LLM generated and modified patch code until the vulnerability is effectively mitigated.

In addition to generating vulnerability patches, LLMs exhibit significant potential for application in the domain of patch porting. Pan *et al.* [58] proposed a solution for automatically porting patches for hard forks using LLM, named PPatHF. They tuned the model with example from the project commit history suitable for training, and adapted the model to the porting patch task by inputting the pre- and post-patch versions of the source project in the hard fork, as well as a specific prompts.

Patch validation is also an important task in the vulnerability repair process, and this task has been facing the dual challenges of inefficiency and high cost for a long time, while the emergence of LLM provides an efficient and low-cost solution

to this task. Zhou *et al.* [59] evaluated the patches generated by automatic program repair using LLM without fine-tuning, utilized in-context learning, and enhanced the model's ability to judge the correctness of the patches by giving the model patch-related information.

LLM-based defence methods mainly prevent attackers from establishing the foothold through vulnerability detection, analysis and repair. Fine-tuning [64], [65] and prompt engineering [66], [67] are currently the mainstream technical methods. Although LLM has shown good application results in these tasks, when dealing with complex and large-scale vulnerabilities, its performance is still significantly limited by the length of the input window.

C. The Defensive Role of LLM in the Lateral Movement Phase

Lateral movement is one of the most critical phases in network attack [68], enabling attackers to escalate privileges, expand system access, exfiltrate sensitive data, or compromise crucial components [69]. Nowadays, the major lateral movement detection methods include IDS, anomaly detection systems, and endpoint detection and response (EDR), which mainly identify potential lateral movement activities by detecting abnormal behaviors, such as unauthorized access attempts, credentials misuse, and abnormal network traffic patterns. However, traditional detection methods rely on rules or signatures, which are difficult to cope with new types of attacks and have limitations in dealing with complex or cross-host behaviors. In contrast, LLM is able to identify anomalous behaviors in lateral movement more accurately through its pattern recognition and inference capabilities, and shows higher flexibility and adaptability, especially in the face of unknown attacks. A comprehensive summary of related works is presented in Table V.

TABLE V
THEMATIC WORKS ABOUT LLMs ON DEFENSE AGAINST LATERAL MOVEMENT ATTACKS.

Detection Task	Detection Data Types	Data Pre-processing methods	Detection Data Analysis Methods	Literature
Intrusion Detection	Vehicle network traffic data	Perform data extraction and preprocessing according to the proposed framework.	Adopt the **reasoning-followed-by-action** pipeline.	[70]
	IoT network traffic data	Use semantic extraction (BSE and USE) and input embedding.	Analyze using the **pre-trained and fine-tuned BERT model**.	[71]
Anomaly Detection	Log data	Encode log entries into vectors and capture sequence information.	Input the **log sequence** into the **trained BERT model** to detect anomalies.	[72]
	Log data	Parsing logs by longest common subsequence and FT-Tree using log parsers.	Use **prompt tuning** to enable PLM to cope with different types of logging anomaly detection.	[73]
EDR	Endpoint data	Converting endpoint data into a structured narrative form endpoint story.	Use the LLM to generate **embeddings** for each text window and detect attack behavior.	[74]

IDS are commonly used to detect lateral movement. Recently, LLMs have demonstrated the ability to perform intrusion detection across various environments, including computer networks, the Internet of Things (IoT), critical infrastructure, and cloud systems, to identify potential lateral movement behaviors [12]. Vehicle network as an emerging network is facing huge cyber threats [75]. To address this issue, Fu *et al.* [71] introduced IoV-BERT-IDS, a BERT-based hybrid IDS for in- and extra-vehicle networks. The system preprocesses traffic data into semantic data suitable for the BERT model through two phases. In fine-tuning, the Unidirectional Semantic Extractor (USE) converts hexadecimal strings into byte sentences by breaking packets into byte units. During pre-training, the Bidirectional Semantic Extractor (BSE) pairs neighboring packets using a sliding window, generating contextual byte sentence pairs. Li *et al.* [70] developed an LLM-based IDS, IDS-Agent, which employs a structured pipeline consisting of inference, action generation, and observation updating to achieve autonomous intrusion detection. It integrates eight action spaces to simplify reasoning and improve decision accuracy. In addition, IDS-Agent utilizes structured long-term memory and external support files for inference.

Anomaly detection based on log files is one of the commonly used ways to detect lateral movement [76]. LLMs leverages its advanced natural language processing capabilities to parse log data and identify anomaly patterns and attack indicators [77], thereby enabling effective lateral movement detection. Huang *et al.* [72] introduced HilBERT, a hierarchical transformer model tailored for system logs. The model utilizes a transformer-based log encoder to vectorize log templates and a transformer-based sequence encoder to integrate these vectors into a unified log sequence representation. Attention mechanisms are employed to capture contextual relationships and derive comprehensive insights across the sequence. Zhang *et al.* [73] proposed LogPrompt, a prompt-based learning framework for log anomaly detection. It uses continuous templates with trainable vectors to adapt to diverse log structures, and a focal loss function to mitigate class imbalance between normal and anomalous logs by focusing on challenging logs.

EDR is another effective method for detecting lateral move-

ment [78]. Portnoy *et al.* [74] introduced a novel methodology for incorporating LLMs into EDR systems to improve the identification of Hands-on-Keyboard network attacks. The researchers converted raw log data into structured "endpoint story" formats and segmented it into smaller windows. A pre-trained LLM was then employed to generate distinct embeddings for each window. These embeddings were subsequently concatenated to create an embedding sequence, which was fed into a training LLM to capture both the global context of the sequence and the inter-window relationships.

LLM-based malicious lateral movement detection is mainly achieved through the analysis of traffic data. Benefiting from LLM's powerful pattern recognition capabilities, LLM has demonstrated excellent performance in security scenarios such as IDS, anomaly detection, and EDR.

D. The Research Gaps in LLM for Post-intrusion Scenarios

Moreover, we note that the majority of current research on LLM-based defense methods primarily addresses competing and preventing network intrusion behaviors. However, a significant research gap exists in the application of LLM-based defense methods to post-intrusion scenarios, including lateral movement, data exfiltration, and post-exfiltration phases. Within the comprehensive defense lifecycle, we contend that post-intrusion network defense is of equal importance. Consequently, it is essential to expand and innovate research on LLM-based defense methods specifically designed for these scenarios, with the goal of fully leveraging the potential of LLMs in tackling post-intrusion challenges.

VI. THE APPLICATION OF LLM IN CTI

CTI can be defined as "evidence-based knowledge, including context, mechanisms, indicators, implications, and actionable advice about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard" [79]. It is an important part of cyber defence, the intelligence foundation of all cyber defence operations, and an important defence operation to make a defender who is in a passive situation
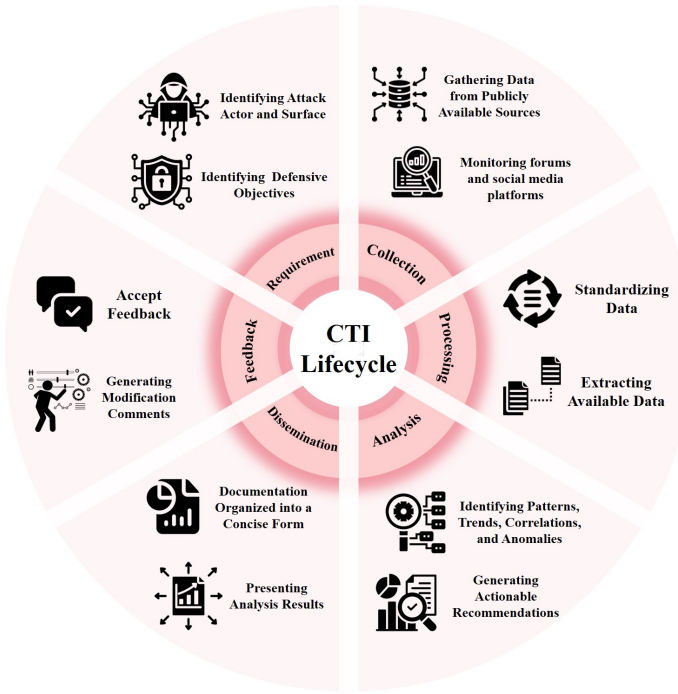
Fig. 3. Lifecycle of CTI.

in a cyber attack become proactive [80]. Due to its important role in cyber defence, we also investigated the application of LLM in this defensive action. Nova [81] divided the CTI lifecycle into six phases: CTI requirements, CTI collection, CTI processing, CTI analysis, CTI dissemination, and CTI feedback, as shown as Fig. 3. LLM is mainly applied in the CTI collection, processing, and analysis phases. In this section, we will introduce the application of LLM in these three phases in detail and all related works in this subsection are summarized in Table VI.

A. LLM for CTI Collection and Processing

The main work in the CTI collection and processing phase is to collect data from traffic logs and publicly available sources and transform them into standardised data format for analysis [81]. In this phase, there is much labour-intensive work, which significantly consumes the energy of network defenders, while LLM can efficiently complete the data collection and standardised processing tasks, which not only effectively reduces the human input but also significantly improves the efficiency of network defence work.

Collecting high-quality original CTI data that meets specific requirements often requires substantial time and effort from security personnel. The advent of LLMs offers a promising solution to mitigate this challenge. Clairoux-Trépanier et al. [82] proposed system, using the GPT-3.5-turbo model to extract CTIs from cybercrime forums. This system updates data daily from cybercrime forums and provides it to the LLM, then uses ten carefully designed prompts to guide the LLM in extracting ten key variables from each forum conversation to describe CTI.

After acquiring the raw data, it must be standardized to facilitate subsequent analysis. However, the original datasets are often unstructured, and traditional manual or rule-based processing methods are not only inefficient but also prone to errors. LLMs provide a novel approach to handle this task. Mitra et al. [83] used LLM to create a modular retrieval-augmented question-answering system, LOCALINTEL. This system generates contextualized local CTI based on the Global CTI Repository and the Local Organizational Database. The system is modular in design, and both the Global CTI Repository and the Global CTI Repository can be replaced to generate CTI that is more relevant to an organisation's needs. Fieblinger et al. [84] proposed a framework to extract CTI from unstructured data sources using LLM automatically. They employed the guidance framework and QLoRA fine-tuning techniques to guide LLMs in extracting CTI triples from unstructured data and subsequently organizing these triples into the structured and queryable Knowledge Graphs. The team found that the guidance framework performed better compared to prompt engineering. Also, incorporating ontology structure and a small number of examples into the prompts improved the quality of the generated CTI triples. CTI View is a BERT-based threat entity identifier that extracts threat entities from cybersecurity report texts [85]. In the CSKG4APT platform proposed by Ren et al. [86], CTI View is optimized to support the processing of cybersecurity reports in English and Chinese. The tool can automate the identification and extraction of threat entity information such as attackers, software/tools, industries, regions, and campaigns from text, providing data support for subsequent CTI analysis.

B. LLM for CTI Analysing

The main task in the CTI analysis phase is identifying patterns, trends, and anomalies from the CTI reports and getting actionable defense recommendations [81]. LLM is equipped with robust information retrieval and data extraction capabilities, which can efficiently extract key information from many reports to assist in the analysis work. At the same time, LLM can use its large knowledge base to provide higher-quality defense recommendations to help network defenders make better defense decisions.

Jin et al. [87] presented Crimson, a system that helps LLMs convert CVE descriptions and CTI information into structured and actionable cyber defense recommendations. Crimson used a Domain-specific Embedding Model to distill complex cyber threat data into a visually and strategically insightful format and Retrieval-Aware Training (RAT) and RAT-R to enable LLMs to use contextually relevant and up-to-date cybersecurity data in their reasoning process. Kucsván et al. [88] developed an automated system for analyzing CTI reports and inferring threat recovery steps using LLMs. By extracting threat behavior triplets from CTI reports, the researchers employed prompt engineering to guide LLMs in deducing appropriate recovery steps. Rahman et al. [89] proposed ChronoCTI, an automated pipeline for mining temporal attack patterns from CTI reports of past cyberattacks. The research team trained Roberta, a fine-tuned version of an existing LLM, using self-constructed sentences-attack techniques mapping dataset so

TABLE VI
THEMATIC WORKS ABOUT LLMS ON CTI.

CTI Phase	Task Objective	Processing Data Format	Approaches	Literature
CTI Collection and Processing	Extracting available CTIs from cyber-crime forums	Forum information	Using **prompts** containing key CTI variables.	[82]
	Generating the organization-specific threat intelligence.	Global threat databases and local knowledge databases data	Using the **modular retrieval-augmented question-answering** system.	[83]
	Extracting CTI triples from unstructured data with enhanced prompt quality.	CTI reports	Using **guidance framework** and **QLoRA fine-tuning** technique.	[84]
	Extracting threat entities from cybersecurity reports.	Unstructured text report collected from various sources	Text extraction and analysis of unstructured cybersecurity reports using the **BERT** model.	[86]
CTI Analysis	Converting CTI into structured and actionable insights.	CVEs and CTIs from diverse sources	Enhancing strategy inference in LLM using **RAT-r fine-tuning** approach.	[87]
	Inferring Threat Recovery Steps from CTI Reports.	CTI reports	Guiding the LLM using the **prompt engineering** technique.	[88]
	Mining temporal attack patterns from CTI reports.	CTI reports	Training the **Roberta model** using self-generated training sets for the task of mining temporal patterns.	[89]

that it can automatically identify and extract attack techniques from CTI reports.

Collecting, processing and analysing CTI was originally a very labour-intensive task. The introduction of LLM has revolutionised this area. The addition of LLM greatly increases the degree of automation in collecting and processing raw CTI data and also providing security personnel with effective assistance when analysing CTI reports. The use of LLM-based CTI technology effectively improves the efficiency of CTI tasks.

VII. APPLICATIONS IN SPECIFIC NETWORK SCENARIOS

In this section, we focus our perspective on the applications of LLM-based network security in different network scenarios, and analyze their main deployment methods and application directions from both traditional and future network perspectives.

A. Traditional Network Scenarios

The traditional network is a static network, usually with a centralized architecture, where the network has limited flexibility and scalability but is widely used. In this network scenario, LLM-based security applications are usually deployed using a centralized deployment approach, and they are generally deployed in the cloud or locally together with other security applications [90]. Most of the current research also adopts this deployment approach by default, for example, the LLM-based detection systems proposed in literature [71] and [72] are deployed on local servers. Although this deployment method is less flexible, it can easily meet the high computational resource requirements of LLM and is also easy to manage. It is also worth noting that current research has been very extensive in exploring the direction of security applications of LLM in traditional network scenarios, covering key aspects such as attack detection, threat analysis, and policy generation, and LLM has demonstrated excellent performance in a number of tasks in real-world tests.

B. Next-Generation Network Scenario

Along with the technological development and demand enhancement, the concept of next-generation network has been put forward. Many types of new networks, such as IoT [91], 6G [92], (Com)^{2}Net [93] and Det(Com)^2 [94], have emerged successively. They are characterized by high dynamics and heterogeneity, as well as multi-layered network structure [95], which brings the convenience of high adaptability, high throughput and low latency, but also brings higher network security risks than traditional networks due to the complexity of its network structure and the diversity of device access [96], [97]. On the other hand, the multi-layered network structure of next-generation network and the limited device resources also bring certain challenges to the deployment and use of traditional network security applications [98], and LLM-based security applications are no exception. However, in recent years, several scholars have conducted in-depth research on this challenge and proposed various solutions.

1) Deployment: The high demand for resources characteristic of LLM makes it difficult to be deployed on resource-limited edge devices in next-generation networks, thus limiting its usage scenarios and application performance [99]. However, techniques such as mixture of experts [100] and federated learning [101] can optimize the model structure and computation to achieve efficient deployment on resource-limited devices. Zhang *et al.* [100] proposed a democratized generative AI framework using compact model strategies, where techniques such as fine-tuning, model pruning, and distillation are used to help LLM deployment on resource-constrained mobile and edge devices. Zhang *et al.* [102] explored the deployment issues in LLM and machine learning (ML) model-driven next-generation networks, and proposed a distributed deployment strategy that deploys LLM and traditional ML models on local or edge base stations (BSs), and each local BS owns a LLM to enhance data privacy and network scalability. Chaoub *et al.* [103] proposed a hybrid deployment strategy for 6G

networks. They integrated lightweight LLM sub-components into network functions (NFs) in the network for real-time tasks, and deployed complex LLM sub-components outside the NFs as standalone microservices to handle complex analysis tasks. The two interact through service-based interfaces or fast APIs. Xu *et al.* [104] proposed a split learning system for 6G networks, which realizes AI services by splitting LLMs into mobile and edge agents. The mobile side runs tiny local LLM, which is responsible for real-time sensing and local interaction, while the edge side runs huge global LLM, which performs complex reasoning and global planning. Both of them collaborate in the network to handle tasks, the mobile side can handle simple tasks independently, while the complex tasks are offloaded to the edge side for processing and then return the results for execution.

2) Application: The new network structure and environment of next-generation network also brings new challenges for network security defense, and the complexity of the network puts forward a new demand for intelligent network management and protection, which LLM happens to have the hope to meet. Li *et al.* [105] proposed an LLM-assisted network operating system framework in which LLM management layer is integrated into the network operating system for strategy generation for service function chain deployment. In this article, the proposed NSGA2-based multi-objective LLM algorithm is innovatively used to find the optimal deployment policy. It effectively improves the intelligence and security of network management. Liu *et al.* [106] pointed out that zero-trust architecture can be used to ensure the security of NGN, the article organizes the zero-trust network through micro-segmentation, and uses the Large Language Model-Enhanced Graph Diffusion (LEGD) algorithm to generate the optimal micro-segmentation, in which LLM is used to generate the dynamic filters based on the information of the network environment to reduce the algorithm search space to improve the algorithm efficiency. The article also proposes a LEGD-Adaptive Maintenance algorithm to respond to trustworthiness updates and service upgrades in the network by fine-tuning the LEGD model. Hong *et al.* [107] proposed a framework for LLM-enabled digital twins networks (DTNs), in which LLMs will be responsible for processing multimodal data in the network. The framework utilizes LLM's own characteristics to enhance data security without affecting the network efficiency, by fine-tuning the way to load sensitive information into the LLM in DTNs, eliminating the data decryption process and reducing the possibility of data leakage, and at the same time, utilizing LLM's reversal curse characteristic [108] to defend against external inference attacks. Satellite-aerial-ground integrated network (SAGIN) is also an important network architecture in NGN [109], due to its heterogeneous, self-organized and dynamic characteristics, which increases the difficulty of network security protection and limits the effectiveness of traditional security methods, and LLM provides a new way to ensure the security of its network [102]. Tang *et al.* [110] pointed out that LLM can significantly enhance the security of SAGIN network through real-time threat detection, automated security policy formulation and dynamic security configuration. Javaid *et al.* [111] also proposed that in similar integrated satellite,

aerial, and terrestrial networks, LLMs can traffic monitoring, malicious behavior identification, and generation of security policies to secure the network, and emphasized that LLMs can guarantee the effectiveness of LLM-enabled security measures through continuous learning.

In traditional network scenarios, LLM-based network security applications have been widely used and demonstrated excellent performance, but in next-generation network scenarios, LLM-based network security applications still face challenges in terms of resources and latency in deployment and use. However, there are now studies that have begun to study techniques such as quantization to optimize the operational efficiency of LLM in new network architectures, explore network security usage scenarios suitable for LLM, and provide feasible solutions for intelligent security protection in next-generation network environments.

VIII. THE SECURITY RISK OF LLM

While LLM protects the network security, it also receives various kinds of network attacks, which may cause the weakening of LLM network defense capability or even threaten the network security of the system where it is located.

A. External Threats

LLM, as a segment of network protection, may also become the target of cyber attackers, who may attack LLM itself and paralyse the LLM-driven defence mechanism to achieve the purpose of successfully hacking into the target system. Currently, various cyber attack threats against LLMs have started to emerge, but as an emerging technology, LLMs are still under-researched in terms of their own defence. This brings great risks to users who use LLM for network defence.

1) Prompt Injection: Among the various attacks, prompt injection are one of the common threats to LLMs [112]. Such attacks enable LLMs to generate undesirable or even malicious output by embedding harmful instructions in the input [113]. For instance, an attacker may conceal malicious prompt content in logs or traffic files, directing LLMs, functioning as IDSs or EDRs, to disregard prior instructions and refrain from alerting users to the attacker's malicious activities. In response to prompt injection attacks, several researchers have proposed mitigation strategies aimed at improving the robustness of LLM against such threats. These strategies aim to ensure that LLM processes input securely, executes only valid commands and resists any potential for hidden malicious content.

Chen *et al.* [114] proposed a structured query approach to defend injection attacks. They used a front-end system to separate the prompt and data parts of the input, encapsulated them into a special data format, and adjusted the structured instructions to allow the LLM to accept inputs encoded in this format, thus allowing the LLM to execute only the commands in the prompt part and not the malicious commands present in the data part. Piet *et al.* [115] propose a method that uses fine-tuning to safeguard LLMs against prompt injection attacks. The research team exploited the fact that LLMs can only execute instructions effectively after specific instruction

tuning. They used either real or self-generated datasets to fine-tune a base LLM (non-instruction-tuned) to focus more on a predefined task. This way, even if the LLM is subjected to a prompt injection attack, it will not execute the malicious instructions.

2) Data Poisoning: On the other hand, data poisoning attacks are also a major threat to LLM. Data poisoning attacks can occur in multiple phases of the LLM life cycle, and attackers can maliciously implant or modify some of the data in the pre-training, fine-tuning, or embedding phases of the model. They implant backdoors and loopholes in the model, which leads to a degradation of the model's performance and may even become the attacker's attack anchors to hack into the target system. In the face of the serious threat posed by data poisoning attacks, researchers aim to enhance the robustness of the model and mitigate the impact of poisoned data on the performance of the model as much as possible through different technical means. In the following, two defence strategies against data poisoning are presented, which propose novel solutions from different perspectives.

Li *et al.* [116] employed the Kullback-Leibler divergence to measure the discrepancy between the probability distributions of the entrusted LLM and a thoroughly cleansed small language model (SLM). Their objective was to minimize the deviation between the output distribution of the integrated model and that of the SLM, thereby achieving k-Near Access-Free. This approach effectively mitigated the influence of poisoned data in the LLM on the final results while maintaining its standard performance. Mo *et al.* [117] proposed a method to mitigate the impact of malicious backdoor attacks on LLMs using a small number of examples. Before user inputs are submitted to the LLM, the research team selects an example from a pool of samples designed to appropriately respond to user requirements. The selected example, which closely matches the current input task, is inserted into the input to guide the model in correctly interpreting the task instruction.

B. Inherent Risks

Currently, LLM still has certain shortcomings in terms of performance. When responding to user needs, LLM may generate biased or incorrect replies, which can negatively impact the performance of applications integrating LLMs, and even allows the current network security system to make incorrect behaviours, which exposes the network system to huge vulnerabilities. This phenomenon is called "misinformation" and constitutes a significant vulnerability when utilizing LLMs as a part of network defense mechanisms. The core reason for generating misinformation lies in the hallucination phenomenon of LLM. Hallucinations occur when LLMs, without truly understanding the content of the training data, use statistical patterns to fill in gaps within the training data, leading to inaccurate or misleading information.

To address the issue of misinformation in LLMs, researchers have proposed corresponding detection methods. Min *et al.* [118] propose FACTSCORE, a framework for assessing the truthfulness of LLM-generated information. The framework subdivides LLM-generated information into atomic facts-units, which are more basic than sentences, and then, utilising the retrieval results from a knowledge source such as Wikipedia, discerns the supportiveness of each atomic facts-units, and ultimately calculates the percentage of supported atomic facts-units as the assessment score.

The hallucination phenomenon may arise due to errors or knowledge gaps in the training data used for the model [119]. To address this issue, higher-quality data can be selected during the training phase to avoid inaccuracies in the dataset. Additionally, external databases can be leveraged by modifying model parameters, injecting up-to-date knowledge, or employing RAG to provide LLM with more comprehensive knowledge. Meanwhile, the inherent limitations of the architecture and training strategies used in LLM may also contribute to the phenomenon of LLM hallucinations. However, the likelihood of LLM hallucinations can be reduced by optimising the way LLMs are trained. Lee *et al.* [120] proposed a Factuality-Enhanced Continued Training approach. In this approach, TOPICPREFIX, representing the topic of each training sample, is added as a prefix to improve the model's understanding of factual information. This prevents information fragmentation caused by document chunking during training. Meanwhile, the zero-masking technique is used for the front part of the sentence, and the loss function is computed only for the latter part of the sentence, which is more prone to errors, to reduce the impact of the entity misassociation problem.

Misinformation may lead to more risky matters, such as improper output handling or excessive agency issues. The improper output handling issue occurs when an erroneous output from an LLM is directly input to a downstream component or system for execution without reasonable validation or processing. In such cases, the defense system may execute incorrect operations or mislead cyber defenders, thereby increasing the likelihood of a successful attack. Conversely, the issue of excessive agency warrants serious attention. Current LLMs can invoke functions or interact with other systems via extensions. Excessive agency privileges in LLMs may pose threats to the security and integrity of network systems when errors occur.

For addressing such extended issues, on one hand, the output of LLMs can undergo secondary validation to ensure that the content is harmless to downstream components and the current system. Alternatively, the output of LLM can be restricted to allow only safe operations. For example, all database operations executed by the LLM can be limited to parameterized queries or prepared statements. On the other hand, minimizing the agency privileges granted to LLM can reduce the impact of errors on the security or integrity of the current system. Adding a "mediator" component is also a feasible solution. The LLM can only send operation requests to the mediator, which determines whether to execute these requests, effectively intercepting unsafe calls.

IX. OPEN PROBLEMS AND RESEARCH DIRECTIONS

Although LLM has been widely used in multiple types of cybersecurity tasks and has achieved excellent results in some of them, there are still many unresolved challenges. Meanwhile, the exploration of LLM in the field of network

security is still in its early stages, and there are still many directions that can be explored. In this section, we provide a brief overview of the open problems and research directions.

A. Open Problems

1) Scarcity of High-Quality Datasets: The current scarcity of high-quality datasets constrains the performance optimisation of LLM on some cybersecurity tasks. On the one hand, due to the changeable forms of current attacks and the emergence of new types of attacks, it leads to the difficulty of data collection [121]. On the other hand, the present training dataset suffers from low data accuracy and high repetitiveness, which affects the training effect and further performance optimisation of LLM [122], [123].

2) Input Length Limitation: The input length limitation also affects the performance of LLM in cybersecurity tasks, especially in the field of vulnerability detection and repair. In the vulnerability detection, most of the existing solutions are limited to function-level detection, and when facing complex vulnerability detection scenarios involving cross-function or cross-class vulnerabilities, LLM is unable to handle complete code fragment information, resulting in a significant degradation of its detection performance [124]. Similarly, in vulnerability repair, the limited context window of LLM cannot accommodate the complete semantic information of a large codebase and the related project background information, which makes it difficult to read enough repair help information, resulting in poor repair results [123].

3) Targeted Attack Threats: LLM is vulnerable to targeted attacks when performing detection and collection of external threat information. For instance, during phishing email detection, the email may contain injection attacks [125]. Or during CTI collection, LLM may be attacked by toxic data due to the complexity of raw intelligence data sources, including unreliable sources such as cybercrime forums and the dark web.

4) Problem of Delayed Inference: In scenarios such as honeypot systems and intrusion detection systems, which have the requirements on system response speed, LLM currently still has the problem of slow real-time response speed, and we think that most of the researches are now mainly focusing on the optimisation of the LLM-base honeypot deception effect and the optimisation of the precision of attack detection, and less consideration is given to the real-time response capability of the LLM-based defence system [126].

5) Black-Box LLMs: Most current studies mostly use black-box LLMs [127], such as the OpenAI GPT series. Although these LLMs have excellent performance, the experimental test samples may have been covered by the pre-training data due to the opacity of the training data, leading to doubts about the reproducibility and reliability of the high repair success rate obtained from the experiments in real scenarios.

B. Research Directions

1) Development of Open-Source and Transparent Datasets: The development of open-source and transparent high-quality datasets can continue to alleviate the current problem of dataset scarcity on the one hand, and on the other hand, mitigate the risk of uncertainty in experimental results due to black-box LLM.

2) Breaking Through the Input Length Limitation: The input length of LLM limits its performance in vulnerability detection and remediation, we think that with the development of LLM technology, the enhancement of the LLM input window capacity may alleviate this problem, on the other hand, current techniques such as code property graph [51] or GNN [52] can also be used to compress the information to alleviate the impact of the input length limitation.

3) Designing Defence Mechanisms Against Pollution Attacks: LLMs frequently interact with unsafe external data in cybersecurity tasks, which may contain pollution inputs such as injection attacks. Such exposure risks degrading model performance or even co-opting LLMs as attack vectors. We think that developing data filtering or isolation techniques to shield LLMs from adversarial contamination is both imperative and a pivotal future research direction.

4) Semantic Data Transformation: As LLM is applied to a wider range of cybersecurity tasks in the future, the data that LLM needs to process will be more complex and diverse, and may not be semantic data suitable for LLM processing. We think that future research needs to allow LLM to cope with more diverse data processing needs, which can be accomplished through data preprocessing, prompt engineering, or domain-specific data training. Future research also focuses on constructing efficient data transformation mechanisms that preserve the critical features and contextual relationships of the original data while reformatting it into information-rich semantic representations.

5) Enhancing Interpretability: Improving the interpretability of LLM analysis results is also an important research direction in the future. In most attack threat detection work, improving interpretability can help security personnel understand the detection results more intuitively, and improve the transparency and trust of the system. It also helps subsequent research to optimise the performance of LLM.

6) Expanding Application Coverage: With the concept of next-generation network, a number of new network architectures have emerged to meet various new demands. While improving network performance, these architectures also bring more attack surfaces and higher security risks, while the complex and changing network structure also increases the difficulty of network protection. LLM with intelligent and adaptive features provides a new solution idea for future network defense. Expanding LLM-based network security solutions from traditional network scenarios to next-generation network scenarios, and studying the deployment and application of network defense solutions in network scenarios such as 6G, IoT, and SAGIN, etc. will be a mainstream research direction in the future.

X. CONCLUSION

In this survey, we not only explore the defensive role of LLM in the various life cycles of cyber attacks, but also point out the obvious research gaps in the post-intrusion

scenario. Through the analysis of relevant literature, we clearly point out the huge application potential of LLM in network security. Although there are already many studies using LLM to accomplish cybersecurity tasks, it should be noted that there are still many unresolved issues and challenges in LLM-based applications. Based on the current status of LLM applications in cybersecurity, we have also listed some possible future research directions. We hope that through this survey, we can provide a systematic thinking and reference framework for future research on the application of LLM in cybersecurity.

REFERENCES

[1] E. Bout, V. Loscri, and A. Gallais, "How machine learning changes the nature of cyberattacks on iot networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 248–279, 2022.

[2] Y. Liu, Y. Li, Y. Wang, X. Zhang, H. B. Gooi, and H. Xin, "Robust and resilient distributed optimal frequency control for microgrids against cyber attacks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 375–386, 2022.

[3] Y. Chen, M. Cui, D. Wang, Y. Cao, P. Yang, B. Jiang, Z. Lu, and B. Liu, "A survey of large language models for cyber threat detection," *Computers & Security*, vol. 145, p. 104016, 2024.

[4] J. Wang, H. Du, Y. Liu, G. Sun, D. Niyato, D. Mao, D. I. Kim, and X. Shen, "Generative ai based secure wireless sensing for isac networks," arXiv preprint arXiv:2408.11398, 2024.

[5] A. D. Syrmakesis, H. H. Alhelou, and N. D. Hatziargyriou, "A novel cyberattack-resilient frequency control method for interconnected power systems using smo-based attack estimation," *IEEE Transactions on Power Systems*, vol. 39, no. 4, pp. 5672–5686, 2024.

[6] X. Kan, Y. Fan, Z. Fang, L. Cao, N. N. Xiong, D. Yang, and X. Li, "A novel iot network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network," *Information Sciences*, vol. 568, pp. 147–162, 2021.

[7] A. Yazdinejad, A. Dehghantanha, R. M. Parizi, G. Srivastava, and H. Karimipour, "Secure intelligent fuzzy blockchain framework: Effective threat detection in iot networks," *Computers in Industry*, vol. 144, p. 103801, 2023.

[8] J. Soikkeli, G. Casale, L. Muñoz-González, and E. C. Lupu, "Redundancy planning for cost efficient resilience to cyber attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1154–1168, 2023.

[9] S. AlDaajeh, H. Saleous, S. Alrabaee, E. Barka, F. Breitinger, and K.-K. Raymond Choo, "The role of national cybersecurity strategies on the improvement of cybersecurity education," *Computers & Security*, vol. 119, p. 102754, 2022.

[10] W. Ding, M. Abdel-Basset, A. M. Ali, and N. Moustafa, "Large language models for cyber resilience: A comprehensive review, challenges, and future perspectives," *Applied Soft Computing*, vol. 170, p. 112663, 2025.

[11] M. Taddeo, T. McCutcheon, and L. Floridi, "Trusting artificial intelligence in cybersecurity is a double-edged sword," *Nature Machine Intelligence*, vol. 1, no. 12, pp. 557–560, 2019.

[12] H. Kheddar, "Transformers and large language models for efficient intrusion detection systems: A comprehensive survey," arXiv preprint arXiv:2408.07583, 2025.

[13] Z. Sheng, Z. Chen, S. Gu, H. Huang, G. Gu, and J. Huang, "Llms in software security: A survey of vulnerability detection techniques and insights," arXiv preprint arXiv:2502.07049, 2025.

[14] X. Zhou, S. Cao, X. Sun, and D. Lo, "Large language model for vulnerability detection and repair: Literature review and the road ahead," *ACM Trans. Softw. Eng. Methodol.*, Dec. 2024, just Accepted.

[15] J. Zhang, H. Bu, H. Wen, Y. Liu, H. Fei, R. Xi, L. Li, Y. Yang, H. Zhu, and D. Meng, "When LLMs meet cybersecurity: a systematic literature review," *Cybersecurity*, vol. 8, no. 1, p. 55, Feb. 2025.

[16] C.-N. Hang, P.-D. Yu, R. Morabito, and C.-W. Tan, "Large language models meet next-generation networking technologies: A review," *Future Internet*, vol. 16, no. 10, 2024.

[17] F. N. Motlagh, M. Hajizadeh, M. Majd, P. Najafi, F. Cheng, and C. Meinel, "Large language models in cybersecurity: State-of-the-art," arXiv preprint arXiv:2402.00891, 2024.

[18] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (llm) security and privacy: The good, the bad, and the ugly," *High-Confidence Computing*, vol. 4, no. 2, p. 100211, 2024.

[19] D. Shu, H. Zhao, X. Liu, D. Demeter, M. Du, and Y. Zhang, "Lawllm: Law large language model for the us legal system," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, ser. CIKM '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 4882–4889.

[20] J. Qiu, K. Lam, G. Li, A. Acharya, T. Y. Wong, A. Darzi, W. Yuan, and E. J. Topol, "Llm-based agentic systems in medicine and healthcare," *Nature Machine Intelligence*, vol. 6, no. 12, pp. 1418–1420, 12 2024.

[21] Q. Wen, J. Liang, C. Sierra, R. Luckin, R. Tong, Z. Liu, P. Cui, and J. Tang, "Ai for education (ai4edu): Advancing personalized education with llm and adaptive learning," ser. KDD '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 6743–6744.

[22] Z. Liu, "Secqa: A concise question-answering dataset for evaluating large language models in computer security," arXiv preprint arXiv:2312.15838, 2023.

[23] N. Tihanyi, M. A. Ferrag, R. Jain, T. Bisztray, and M. Debbah, "Cybermetric: A benchmark dataset based on retrieval-augmented generation for evaluating llms in cybersecurity knowledge," in *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2024, pp. 296–302.

[24] Z. Liu, J. Shi, and J. F. Buford, "Cyberbench: A multi-task benchmark for evaluating large language models in cybersecurity," AAAI-24 Workshop on Artificial Intelligence for Cyber Security (AICS), 2024.

[25] D. Bhusal, M. T. Alam, L. Nguyen, A. Mahara, Z. Lightcap, R. Frazier, R. Fieblinger, G. L. Torales, B. A. Blakely, and N. Rastogi, "Secure: Benchmarking large language models for cybersecurity," arXiv preprint arXiv:2405.20441, 2024.

[26] J. Zhang, H. Wen, L. Deng, M. Xin, Z. Li, L. Li, H. Zhu, and L. Sun, "Hackmentor: Fine-tuning large language models for cybersecurity," in *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2023, pp. 452–461.

[27] G. Siracusano, D. Sanvito, R. Gonzalez, M. Srinivasan, S. Kamatchi, W. Takahashi, M. Kawakita, T. Kakumaru, and R. Bifulco, "Time for action: Automated analysis of cyber threat intelligence in the wild," arXiv preprint arXiv:2307.10214, 2023.

[28] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019.

[29] S. Sebastián, R.-G. Diugan, J. Caballero, I. Sanchez-Rola, and L. Bilge, "Domain and website attribution beyond whois," in *Proceedings of the 39th Annual Computer Security Applications Conference*, ser. ACSAC '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 124–137.

[30] V. R. Saraswathi, I. S. Ahmed, S. M. Reddy, S. Akshay, V. M. Reddy, and S. M. Reddy, "Automation of recon process for ethical hackers," in *2022 International Conference for Advancement in Technology (ICONAT)*, 2022, pp. 1–6.

[31] G. Tian, C. Zhang, A. M. Fatollahi-Fard, Z. Li, C. Zhang, and Z. Jiang, "An enhanced social engineering optimizer for solving an energy-efficient disassembly line balancing problem based on bucket brigades and cloud theory," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 5, pp. 7148–7159, 2023.

[32] S. Roy, N. Sharmin, J. C. Acosta, C. Kiekintveld, and A. Laszka, "Survey and taxonomy of adversarial reconnaissance techniques," *ACM Comput. Surv.*, vol. 55, no. 6, Dec. 2022.

[33] S. Wang, Q. Pei, Y. Zhang, X. Liu, and G. Tang, "A hybrid cyber defense mechanism to mitigate the persistent scan and foothold attack," *Security and Communication Networks*, vol. 2020, no. 1, p. 8882200, 2020.

[34] H. Bijmans, T. Booij, A. Schwedersky, A. Nedgabat, and R. van Wegberg, "Catching phishers by their bait: Investigating the dutch phishing landscape through phishing kit detection," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 3757–3774.

[35] J. Gan, C. Luo, W. Shi, Y. Liu, X. Liu, and Z. Tian, "An attack exploiting cyber-arm industry," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 2, pp. 1686–1702, 2025.

[36] N. Mohamed and B. Belaton, "Sbi model for the detection of advanced persistent threat based on strange behavior of using credential dumping technique," *IEEE Access*, vol. 9, pp. 42 919–42 932, 2021.

[37] M. A. R. A. Amin, S. Shetty, L. Njilla, D. K. Tosh, and C. Kamhoua, "Hidden markov model and cyber deception for the prevention of adversarial lateral movement," *IEEE Access*, vol. 9, pp. 49662–49682, 2021.

[38] A. Sharma, B. B. Gupta, A. K. Singh, and V. Saraswat, "Advanced persistent threats (apt): evolution, anatomy, attribution and countermeasures," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 7, pp. 9355–9381, 2023.

[39] T. Zhang, C. Xu, J. Shen, X. Kuang, and L. A. Grieco, "How to disturb network reconnaissance: A moving target defense approach based on deep reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5735–5748, 2023.

[40] M. Hassanin, M. Keshk, S. Salim, M. Alsubaie, and D. Sharma, "Pllmcs: Pre-trained large language model (llm) for cyber threat detection in satellite networks," *Ad Hoc Networks*, vol. 166, p. 103645, 2025.

[41] C. Song, L. Ma, J. Zheng, J. Liao, H. Kuang, and L. Yang, "Audit-llm: Multi-agent collaboration for log-based insider threat detection," arXiv preprint arXiv:2408.08902, 2024.

[42] M. Sladić, V. Valeros, C. Catania, and S. Garcia, "Llm in the shell: Generative honeypots," in *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2024, pp. 430–435.

[43] L. Li and B. Gong, "Prompting large language models for malicious webpage detection," in *2023 IEEE 4th International Conference on Pattern Recognition and Machine Learning (PRML)*, 2023, pp. 393–400.

[44] T. Koide, N. Fukushi, H. Nakano, and D. Chiba, "Chatspamdetector: Leveraging large language models for effective phishing email detection," arXiv preprint arXiv:2402.18093, 2024.

[45] C. Fang, N. Miao, S. Srivastav, J. Liu, R. Zhang, R. Fang, Asmita, R. Tsang, N. Nazari, H. Wang, and H. Homayoun, "Large language models for code analysis: Do llms really do their job?" in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 829–846.

[46] K. Stein, A. A. Mahyari, G. Francia, and E. El-Sheikh, "Towards novel malicious packet recognition: A few-shot learning approach," in *MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM)*, 2024, pp. 847–852.

[47] Z. Yu, M. Wen, X. Guo, and H. Jin, "Maltracker: A fine-grained npm malware tracker copiloted by llm-enhanced dataset," in *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, ser. ISSTA 2024. New York, NY, USA: Association for Computing Machinery, 2024, p. 1759–1771.

[48] C. Chhetri, "Exploring large language model-powered pedagogical approaches to cybersecurity education," in *Proceedings of the 25th Annual Conference on Information Technology Education*, ser. SIGITE '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 163–166.

[49] H. İŞ, "Llm-driven sat impact on phishing defense: A cross-sectional analysis," in *2024 12th International Symposium on Digital Forensics and Security (ISDFS)*, 2024, pp. 1–5.

[50] S. Wang, Q. Pei, Y. Xiao, F. Shao, S. Yuan, J. Chu, and R. Liao, "Probabilistic models for evaluating network edge's resistance against scan and foothold attack," *IET Communications*, vol. 18, no. 20, pp. 1983–1995, 2024.

[51] G. Lu, X. Ju, X. Chen, W. Pei, and Z. Cai, "Grace: Empowering llm-based software vulnerability detection with graph structure and in-context learning," *Journal of Systems and Software*, vol. 212, p. 112031, 2024.

[52] A. Z. H. Yang, H. Tian, H. Ye, R. Martins, and C. L. Goues, "Security vulnerability detection with multitask self-instructed fine-tuning of large language models," arXiv preprint arXiv:2406.05892, 2024.

[53] C. Zhang, L. Wang, D. Fan, J. Zhu, T. Zhou, L. Zeng, and Z. Li, "Vtt-llm: Advancing vulnerability-to-tactic-and-technique mapping through fine-tuning of large language model," *Mathematics*, vol. 12, no. 9, p. 1286, 2024.

[54] J. Yin, M. Tang, J. Cao, and H. Wang, "Apply transfer learning to cybersecurity: Predicting exploitability of vulnerabilities by description," *Knowledge-Based Systems*, vol. 210, p. 106529, 2020.

[55] Y. Luo, W. Xu, K. Andersson, M. S. Hossain, and D. Xu, "Fellmvp: An ensemble llm framework for classifying smart contract vulnerabilities," in *2024 IEEE International Conference on Blockchain (Blockchain)*, 2024, pp. 89–96.

[56] C. Wang, J. Zhang, J. Gao, L. Xia, Z. Guan, and Z. Chen, "Contracttinker: Llm-empowered vulnerability repair for real-world smart contracts," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 2350–2353.

[57] M. C. Tol and B. Sunar, "Zeroleak: Automated side-channel patching in source code using llms," in *Computer Security – ESORICS 2024*, J. Garcia-Alfaro, R. Kozik, M. Choraś, and S. Katsikas, Eds. Cham: Springer Nature Switzerland, 2024, pp. 290–310.

[58] S. Pan, Y. Wang, Z. Liu, X. Hu, X. Xia, and S. Li, "Automating zero-shot patch porting for hard forks," in *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, ser. ISSTA 2024. New York, NY, USA: Association for Computing Machinery, 2024, p. 363–375.

[59] X. Zhou, B. Xu, K. Kim, D. Han, H. H. Nguyen, T. Le-Cong, J. He, B. Le, and D. Lo, "Leveraging large language model for automatic patch correctness assessment," *IEEE Transactions on Software Engineering*, vol. 50, no. 11, pp. 2865–2883, 2024.

[60] A. Happe and J. Cito, "Getting pwn'd by ai: Penetration testing with large language models," in *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2023. New York, NY, USA: Association for Computing Machinery, 2023, p. 2082–2086.

[61] X. Yin, C. Ni, and S. Wang, "Multitask-based evaluation of open-source llm on software vulnerability," *IEEE Transactions on Software Engineering*, vol. 50, no. 11, pp. 3071–3087, 2024.

[62] J. Zhang, C. Wang, A. Li, W. Sun, C. Zhang, W. Ma, and Y. Liu, "An empirical study of automated vulnerability localization with large language models," arXiv preprint arXiv:2404.00287, 2024.

[63] J. Wichelmann, A. Moghimi, T. Eisenbarth, and B. Sunar, "Microwalk: A framework for finding side channels in binaries," in *Proceedings of the 34th Annual Computer Security Applications Conference*, ser. ACSAC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 161–173.

[64] Y. Chen, Z. Ding, L. Alowain, X. Chen, and D. Wagner, "Diversevul: A new vulnerable source code dataset for deep learning based vulnerability detection," in *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, ser. RAID '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 654–668.

[65] Y. Guo, C. Patsakis, Q. Hu, Q. Tang, and F. Casino, "Outside the comfort zone: Analysing llm capabilities in software vulnerability detection," in *Computer Security – ESORICS 2024*, J. Garcia-Alfaro, R. Kozik, M. Choraś, and S. Katsikas, Eds. Cham: Springer Nature Switzerland, 2024, pp. 271–289.

[66] M. Fu, C. K. Tantithamthavorn, V. Nguyen, and T. Le, "Chatgpt for vulnerability detection, classification, and repair: How far are we?" in *2023 30th Asia-Pacific Software Engineering Conference (APSEC)*, 2023, pp. 632–636.

[67] Y. Nong, M. Aldeen, L. Cheng, H. Hu, F. Chen, and H. Cai, "Chain-of-thought prompting of large language models for discovering and fixing software vulnerabilities," arXiv preprint arXiv:2402.17230, 2024.

[68] Y. Fang, C. Wang, Z. Fang, and C. Huang, "Lmtracker: Lateral movement path detection based on heterogeneous graph embedding," *Neurocomputing*, vol. 474, pp. 37–47, 2022.

[69] J. Khoury, D. Klisura, H. Zanddizari, G. De La Torre Parra, P. Najafirad, and E. Bou-Harb, "Jbeil: Temporal graph-based inductive learning to infer lateral movement in evolving enterprise networks," in *2024 IEEE Symposium on Security and Privacy (SP)*, 2024, pp. 3644–3660.

[70] Y. Li, Z. Xiang, N. D. Bastian, D. Song, and B. Li, "IDS-agent: An LLM agent for explainable intrusion detection in iot networks," in *NeurIPS 2024 Workshop on Open-World Agents*, 2024.

[71] M. Fu, P. Wang, M. Liu, Z. Zhang, and X. Zhou, "Iov-bert-ids: Hybrid network intrusion detection system in iov using large language models," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 2, pp. 1909–1921, 2025.

[72] S. Huang, Y. Liu, C. Fung, H. Wang, H. Yang, and Z. Luan, "Improving log-based anomaly detection by pre-training hierarchical transformers," *IEEE Transactions on Computers*, vol. 72, no. 9, pp. 2656–2667, 2023.

[73] T. Zhang, X. Huang, W. Zhao, S. Bian, and P. Du, "Logprompt: A log-based anomaly detection framework using prompts," in *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023, pp. 1–8.

[74] A. Portnoy, E. Azikri, and S. Kels, "Towards automatic hands-on-keyboard attack detection using llms in edr solutions," arXiv preprint arXiv:2408.01993, 2024.

[75] T. Zhang, C. Xu, P. Zou, H. Tian, X. Kuang, S. Yang, L. Zhong, and D. Niyato, "How to mitigate ddos intelligently in sd-iov: A moving target defense approach," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 1097–1106, 2023.

[76] H. Bian, T. Bai, M. A. Salahuddin, N. Limam, A. A. Daya, and R. Boutaba, "Uncovering lateral movement using authentication logs," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 1049–1063, 2021.

[77] V.-H. Le and H. Zhang, "Log parsing: How far can chatgpt go?" in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2023, pp. 1699–1704.

[78] C. Smiliotopoulos, G. Kambourakis, and C. Kolias, "Detecting lateral movement: A systematic survey," *Heliyon*, vol. 10, no. 4, p. e26317, 02 2024, doi: 10.1016/j.heliyon.2024.e26317.

[79] R. McMillan, "Definition: Threat intelligence," [Online], 2022, accessed: Dec. 10, 2024. [Online]. Available: https://gartner.com/

[80] N. Sun, M. Ding, J. Jiang, W. Xu, X. Mo, Y. Tai, and J. Zhang, "Cyber threat intelligence mining for proactive cybersecurity defense: a survey and new perspectives," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 3, pp. 1748–1774, 2023.

[81] K. Nova, "Security and resilience in sustainable smart cities through cyber threat intelligence," *International Journal of Information and Cybersecurity*, vol. 6, no. 1, pp. 21–42, 2022.

[82] V. Clairoux-Trepanier, I.-M. Beauchamp, E. Ruellan, M. Paquet-Clouston, S.-O. Paquette, and E. Clay, "The use of large language models (llm) for cyber threat intelligence (cti) in cybercrime forums," *arXiv preprint arXiv:2408.03354*, 2024.

[83] S. Mitra, S. Neupane, T. Chakraborty, S. Mittal, A. Piplai, M. Gaur, and S. Rahimi, "Localintel: Generating organizational threat intelligence from global and local cyber knowledge," *arXiv preprint arXiv:2401.10036*, 2024.

[84] R. Fieblinger, M. T. Alam, and N. Rastogi, "Actionable cyber threat intelligence using knowledge graphs and large language models," in *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2024, pp. 100–111.

[85] Y. Zhou, Y. Tang, M. Yi, C. Xi, and H. Lu, "Cti view: Apt threat intelligence analysis system," *Security and Communication Networks*, vol. 2022, no. 1, p. 9875199, 2022.

[86] Y. Ren, Y. Xiao, Y. Zhou, Z. Zhang, and Z. Tian, "Cskg4apt: A cyber-security knowledge graph for advanced persistent threat organization attribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 5695–5709, 2023.

[87] J. Jin, B. Tang, M. Ma, X. Liu, Y. Wang, Q. Lai, J. Yang, and C. Zhou, "Crimson: Empowering strategic reasoning in cybersecurity through large language models," arXiv preprint arXiv:2403.00878, 2024.

[88] Z. L. Kucsván, M. Caselli, A. Peter, and A. Continella, "Inferring recovery steps from cyber threat intelligence reports," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, F. Maggi, M. Egele, M. Payer, and M. Carminati, Eds. Cham: Springer Nature Switzerland, 2024, pp. 330–349.

[89] M. R. Rahman, B. Wroblewski, Q. Matthews, B. Morgan, T. Menzies, and L. Williams, "Mining temporal attack patterns from cyberthreat intelligence reports," arXiv preprint arXiv:2401.01883, 2024.

[90] Y. Shen, J. Shao, X. Zhang, Z. Lin, H. Pan, D. Li, J. Zhang, and K. B. Letaief, "Large language models empowered autonomous edge ai for connected intelligence," *IEEE Communications Magazine*, vol. 62, no. 10, pp. 140–146, 2024.

[91] H. R. Chi, C. K. Wu, N.-F. Huang, K.-F. Tsang, and A. Radwan, "A survey of network automation for industrial internet-of-things toward industry 5.0," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 2065–2077, 2023.

[92] L.-H. Shen, K.-T. Feng, and L. Hanzo, "Five facets of 6g: Research challenges and opportunities," *ACM Comput. Surv.*, vol. 55, no. 11, Feb. 2023.

[93] W. Zhang, D. Yang, C. Zhang, Q. Ye, H. Zhang, and X. Shen, "(com)2net: A novel communication and computation integrated network architecture," *IEEE Network*, vol. 38, no. 2, pp. 35–44, 2024.

[94] W. Zhang, N. Tang, D. Yang, R. Guo, H. Zhang, and X. Shen, "Det(com)2: Deterministic communication and computation integration toward aigc services," *IEEE Wireless Communications*, vol. 31, no. 3, pp. 32–41, 2024.

[95] W. Zhang, Y. He, T. Zhang, C. Ying, and J. Kang, "Intelligent resource adaptation for diversified service requirements in industrial iot," *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2024.

[96] K. Sood, M. R. Nosouhi, D. D. N. Nguyen, F. Jiang, M. Chowdhury, and R. Doss, "Intrusion detection scheme with dimensionality reduction in next generation networks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 965–979, 2023.

[97] X. Zhu, J. Liu, L. Lu, T. Zhang, T. Qiu, C. Wang, and Y. Liu, "Enabling intelligent connectivity: A survey of secure isac in 6g networks," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2024.

[98] T. Zhang, F. Kong, D. Deng, X. Tang, X. Wu, C. Xu, L. Zhu, J. Liu, B. Ai, Z. Han, and R. H. Deng, "Moving target defense meets artificial intelligence-driven network: A comprehensive survey," *IEEE Internet of Things Journal*, pp. 1–1, 2025.

[99] G. Qu, Q. Chen, W. Wei, Z. Lin, X. Chen, and K. Huang, "Mobile edge intelligence for large language models: A contemporary survey," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2025.

[100] R. Zhang, J. He, X. Luo, D. Niyato, J. Kang, Z. Xiong, Y. Li, and B. Sikdar, "Toward democratized generative ai in next-generation mobile edge networks," *IEEE Network*, pp. 1–1, 2025.

[101] Y. Liu, J. Yin, W. Zhang, C. An, Y. Xia, and H. Zhang, "Integration of federated learning and ai-generated content: A survey of overview, opportunities, challenges, and solutions," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2024.

[102] R. Zhang, H. Du, D. Niyato, J. Kang, Z. Xiong, A. Jamalipour, P. Zhang, and D. I. Kim, "Generative ai for space-air-ground integrated networks," *IEEE Wireless Communications*, vol. 31, no. 6, pp. 10–20, 2024.

[103] A. Chaoub and M. Elkotob, "Mobile network-specialized large language models for 6g: Architectures, innovations, challenges, and future trends," arXiv preprint arXiv:2502.04933, 2025.

[104] M. Xu, D. Niyato, J. Kang, Z. Xiong, S. Mao, Z. Han, D. I. Kim, and K. B. Letaief, "When large language model agents meet 6g networks: Perception, grounding, and alignment," *IEEE Wireless Communications*, vol. 31, no. 6, pp. 63–71, 2024.

[105] Y. Li, Q. Zhang, H. Yao, R. Gao, X. Xin, and M. Guizani, "Next-gen service function chain deployment: Combining multi-objective optimization with ai large language models," *IEEE Network*, pp. 1–1, 2025.

[106] Y. Liu, G. Liu, H. Du, D. Niyato, J. Kang, Z. Xiong, D. I. Kim, and X. Shen, "Hierarchical micro-segmentations for zero-trust services via large language model (llm)-enhanced graph diffusion," arXiv preprint arXiv:2406.13964, 2024.

[107] Y. Hong, J. Wu, and R. Morello, "Llm-twin: mini-giant model-driven beyond 5g digital twin networking framework with semantic secure communication and computation," *Scientific Reports*, vol. 14, no. 1, p. 19065, 08 2024.

[108] Z. Lu, L. Jin, P. Li, Y. Tian, L. Zhang, S. Wang, G. Xu, C. Tian, and X. Cai, "Rethinking the reversal curse of LLMs: a prescription from human knowledge reversal," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 7518–7530.

[109] H. KONG, M. LIN, J. ZHANG, J. OUYANG, J.-B. WANG, and P. K. UPADHYAY, "Ergodic sum rate for uplink noma transmission in satellite-aerial-ground integrated networks," *Chinese Journal of Aeronautics*, vol. 35, no. 9, pp. 58–70, 2022.

[110] J. Tang, F. Tang, S. Long, M. Zhao, and N. Kato, "Utilizing large language models for advanced optimization and intelligent management in space-air-ground integrated networks," *IEEE Network*, pp. 1–1, 2024.

[111] S. Javaid, R. A. Khalil, N. Saeed, B. He, and M.-S. Alouini, "Leveraging large language models for integrated satellite-aerial-terrestrial networks: Recent advances and future directions," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 399–432, 2025.

[112] Y. Liu, Y. Jia, R. Geng, J. Jia, and N. Z. Gong, "Formalizing and benchmarking prompt injection attacks and defenses," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 1831–1847.

[113] X. Huang, W. Ruan, W. Huang, G. Jin, Y. Dong, C. Wu, S. Bensalem, R. Mu, Y. Qi, X. Zhao *et al.*, "A survey of safety and trustworthiness of large language models through the lens of verification and validation," *Artificial Intelligence Review*, vol. 57, no. 7, p. 175, 2024.

[114] S. Chen, J. Piet, C. Sitawarin, and D. Wagner, "Struq: Defending against prompt injection with structured queries," 2024.

[115] J. Piet, M. Alrashed, C. Sitawarin, S. Chen, Z. Wei, E. Sun, B. Alomair, and D. Wagner, "Jatmo: Prompt injection defense by task-specific finetuning," in *Computer Security – ESORICS 2024*, J. Garcia-Alfaro, R. Kozik, M. Choraś, and S. Katsikas, Eds. Cham: Springer Nature Switzerland, 2024, pp. 105–124.

[116] T. Li, Q. Liu, T. Pang, C. Du, Q. Guo, Y. Liu, and M. Lin, "Purifying large language models by ensembling a small language model," arXiv preprint arXiv:2402.14845, 2024.

[117] W. Mo, J. Xu, Q. Liu, J. Wang, J. Yan, H. Askari, C. Xiao, and M. Chen, "Test-time backdoor mitigation for black-box large

language models with defensive demonstrations," arXiv preprint arXiv:2311.09763, 2025.

[118] S. Min, K. Krishna, X. Lyu, M. Lewis, W. tau Yih, P. W. Koh, M. Iyyer, L. Zettlemoyer, and H. Hajishirzi, "Factscore: Fine-grained atomic evaluation of factual precision in long form text generation," arXiv preprint arXiv:2305.14251, 2023.

[119] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–55, 2025.

[120] N. Lee, W. Ping, P. Xu, M. Patwary, P. N. Fung, M. Shoeybi, and B. Catanzaro, "Factuality enhanced language models for open-ended text generation," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 34 586–34 599.

[121] Y. Xu, Q. Zhang, H. Deng, Z. Liu, C. Yang, and Y. Fang, "Unknown web attack threat detection based on large language model," *Applied Soft Computing*, vol. 173, p. 112905, 2025.

[122] R. Croft, M. A. Babar, and M. M. Kholoosi, "Data quality for software vulnerability datasets," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, 2023, pp. 121–133.

[123] U. Kulsum, H. Zhu, B. Xu, and M. d'Amorim, "A case study of llm for automated vulnerability repair: Assessing impact of reasoning and patch validation feedback," in *Proceedings of the 1st ACM International Conference on AI-Powered Software*, ser. AIware 2024. New York, NY, USA: Association for Computing Machinery, 2024, p. 103–111.

[124] X. Zhou, D.-M. Tran, T. Le-Cong, T. Zhang, I. C. Irsan, J. Sumarlin, B. Le, and D. Lo, "Comparison of static application security testing tools and large language models for repo-level vulnerability detection," arXiv preprint arXiv:2407.16235, 2024.

[125] T. Koide, H. Nakano, and D. Chiba, "Chatphishdetector: Detecting phishing sites using large language models," *IEEE Access*, vol. 12, pp. 154 381–154 400, 2024.

[126] H. V. Vo, H. P. Du, and H. N. Nguyen, "Apelid: Enhancing real-time intrusion detection with augmented wgan and parallel ensemble learning," *Computers & Security*, vol. 136, p. 103567, 2024.

[127] H. Pearce, B. Tan, B. Ahmad, R. Karri, and B. Dolan-Gavitt, "Examining zero-shot vulnerability repair with large language models," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 2339–2356.

Tao Zhang (Member, IEEE) received the B.S. degree in Internet of Things engineering from the Beijing University of Posts and Telecommunications (BUPT) and the Queen Mary University of London in 2018, and the Ph.D. degree in computer science and technology from BUPT in 2023. He is currently an Associate Professor with the School of Cyberspace Science and Technology, Beijing Jiaotong University. His publications include ESI highly cited paper and well-archived international journals and proceedings, such as IEEE COMST, JSAC, TIFS, TDSC, TMC, TITS, TCCN and TII etc. His research interests include network security, moving target defense, and federated learning. He has served as the guest editor for Electronics and Chinese Journal of Network and Information Security, and the TPC chair and a PC member for some international conferences and workshops. He was a recipient of the Best Paper Award from NaNA 2018, IWCMC 2021, DIONE 2024, and ICA3PP 2024, and a recipient of Outstanding Paper Award from iThings 2023, and SmartCity 2024. His Ph.D. thesis was awarded the Outstanding Doctoral Dissertation by BUPT in 2023.

Jiqiang Liu (Senior Member, IEEE) received the Ph.D. degree from Beijing Normal University in 1999. He is currently a Full Professor and the Dean of the School of Cyberspace Science and Technology, Beijing Jiaotong University. He has authored or coauthored over 200 publications. In recent years, he has been mainly engaged in research on trusted computing, privacy protection, and cloud computing security.

Jiacheng Wang is the postdoctoral research fellow in the College of Computing and Data Science, Nanyang Technological University, Singapore. Prior to that, he received the Ph.D. degree in School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include wireless sensing, generative artificial intelligence, and semantic communications

Xuangou Wu received the Ph.D. degree from the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China, in 2013. He is currently a Full Professor and the Dean of the School of Computer Science and Technology, University of Anhui Technology, Maanshan, China. His research interests include Intelligent Internet of Things, Network Security, and Privacy Protection.

Shuang Tian received the B.Eng. degree in computer science and technology from China University of Geosciences Beijing, Beijing, China, in 2024. He is currently working toward the M.Eng. degree with the School of Software Engineering, Beijing Jiaotong University, Beijing, China.

Xiaoqiang Zhu (M'23) received the Ph.D. degree in software engineering from Tianjin University, China, in 2022, and the M.S. degree in computer science from Dalian University of Technology, China, in 2018. He served as a joint Ph.D. student at ETH Zurich, Switzerland, supported by the China Scholarship Council in 2021. He is currently an Assistant Professor (Lecturer) with the School of Cyberspace Science and Technology, Beijing Jiaotong University, China. He has published scientific papers in international journals, such as IEEE COMST, TMC, TNSE, etc..

Dong In Kim (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1990. He is currently a Distinguished Professor with the College of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea. He is a Fellow of the Korean Academy of Science and Technology and a Life Member of the National Academy of Engineering of Korea. He received several research awards, including the 2023 IEEE ComSoc Best Survey Paper Award and the 2022 IEEE Best Land Transportation Paper Award.

Ruichen Zhang (Member, IEEE) is currently working as a PostDoctoral Research Fellow with the College of Computing and Data Science, Nanyang Technological University (NTU), Singapore. He received the B.E. degree from Henan University (HENU), China, in 2018, and the Ph.D. degree from Beijing Jiaotong University (BJTU), China, in 2023. In 2024, he was a Visiting Scholar with the College of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea. His research interests include LLM-empowered networking, reinforcement learning-enabled wireless communication, generative AI models, and heterogeneous networks.

Weiting Zhang (Member, IEEE) received the PhD degree in communication and information systems with Beijing Jiaotong University, Beijing, China, in 2021. From 2019 to 2020, he was a visiting PhD student with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Starting from December 2021, he works as an associate professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include industrial Internet of Things, federated learning and edge intelligence.

Zhenhui Yuan (Senior Member, IEEE) received the B.Eng. degree in software engineering with Wuhan University, Wuhan, China, in 2008, and the Ph.D. degree in electronic engineering from Dublin City University, Dublin, Ireland, in 2012. He is currently an Assistant Professor with the University of Warwick. He was the Lead Guest Editor in IEEE NETWORK and IEEE INTERNET OF THINGS JOURNAL. He is also the founding Chair of VeSUS (6G-empowered Robotic Vehicles for Sustainable Development) Workshop.

Shiwen Mao (Fellow, IEEE) is a Professor and Earle C. Williams Eminent Scholar and Director of the Wireless Engineering Research and Education Center at Auburn University. Dr. Mao's research interest includes wireless networks, multimedia communications, and smart grid. He is the editor-in-chief of IEEE Transactions on Cognitive Communications and Networking and a member-atlarge on the Board of Governors of IEEE Communications Society. He received the IEEE ComSoc MMTC Outstanding Researcher Award in 2023, and the SEC 2023 Faculty Achievement Award for Auburn.