This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3510936

Large Language Models for Networking: Workflow, Advances and Challenges

Chang Liu, Xiaohui Xie, Xinggong Zhang *Senior Member, IEEE*, Yong Cui, *Member, IEEE*

***Abstract*—The networking domain is characterized by its high complexity and rapid iteration, requiring extensive expertise to accomplish network tasks, ranging from network design, configuration, diagnosis and security. The inherent complexity of these tasks, coupled with the ever-changing landscape of networking technologies and protocols, poses significant hurdles for traditional methods. However, the recent emergence of large language models (LLMs) has sparked a new wave of possibilities in addressing these challenges. LLMs have demonstrated remarkable capabilities in natural language understanding, generation, and reasoning. These models, trained on extensive data, can benefit the networking domain. Some efforts have already explored the application of LLMs in the networking domain and revealed promising results. To describe the fundamental process involved in applying LLM for networking, we first propose a unified workflow that encompasses a majority of the previous work. We then introduce the highlights of existing works by category and explain in detail how they operate at different stages of the workflow. Furthermore, we delve into the challenges encountered, discuss potential solutions, and outline future research prospects. We hope that this survey will provide insight for researchers and practitioners, promoting the development of this interdisciplinary research field.**

***Index Terms*—Large Language Models, Networking, LLMN Workflow**

I. Introduction

Networks, including wide-area internet, data center networks, and satellite networks, are critical infrastructures with distinct design principles and architectures. Due to the complexity and rapid evolution in the networking domain, managing and maintaining these networks involves complex operations that demand extensive network expertise. Efficient solutions for accomplishing these tasks are essential for ensuring network stability and scalability, which have garnered significant attention from both academia and industry.

Although several machine learning (ML)-based methods have been proposed and shown to be effective, they often lack generalization and necessitate repeated designs for different scenarios. Recently, there has been significant progress in large language models (LLMs), representing the latest advancement in generative artificial intelligence (GAI). Pre-trained on extensive data, these models exhibit remarkable capabilities in concept understanding, logical reasoning, and tool usage, driving groundbreaking advancements in various domains, such as chip design, protein structure generation, and robot-embodied intelligence. Inspired by these promising endeavors, some researchers have begun exploring the application of LLMs for networking, demonstrating encouraging results. In particular, LLMs offer significant advantages in networking. LLMs reduce task workloads by eliminating the need for strict data modeling and feature extraction. They can also construct logical chains for complex problem-solving by leveraging domain knowledge. Additionally, their transfer learning capability allows them to apply insights from related fields to new network challenges, enhancing their flexibility and applicability. Furthermore, due to their ability to process natural language input and output, LLMs can utilize various tools to automate task execution.

Given the growing interest in large language models for networking (LLMN), we believe that it is the right time to survey the current literature. Based on the in-depth analysis of existing studies, we first propose an abstract and unified workflow to describe the fundamental processes involved in LLMN. The LLMN workflow (Section III)consists of six stages: task definition, data representation, prompt engineering, model evolution, tools integration, and validation. These stages aim to address complex tasks, handle diverse data types, guide LLMs in generating accurate answers, expand LLM capabilities through tool integration, and ensure the reliability of the execution results.

We then conduct a targeted review (Section IV) of notable advancements in networking facilitated by LLMs. For example, LLMs can assist in designing algorithms and managing network topologies [1], [2], enhancing configuration efficiency while minimizing errors [3], [4], identifying and eliminating hidden patterns or anomalies [5], [6], and introducing novel strategies for improving network security [7] [8]. We categorize these advancements into several key fields of networking, including network design, network configuration, network diagnosis, and network security, and elaborate on how they are performed at each stage of the LLMN workflow.

In addition to the literature review, we also identify several challenges (Section V) that necessitate further research. These challenges span across different stages of the LLMN workflow and can be categorized into the following aspects: possessing intelligent planning capabilities, understanding network traffic, prompt engineering for deliberate reasoning, developing network-specific LLMs, utilizing tools autonomously, and ensuring reliability and safety through a validation environment. Potential approaches to address these challenges are also discussed.

Previous studies have also attempted to propose paradigms for the application of LLMs in the networking domain. For

C. Liu, X. Xie and Y. Cui are with the Department of Computer Science and Technology, Tsinghua University, Beijing, China.

X. Zhang is with the Wangxuan Institute of Computer Technology, Peking University, Beijing, China.

Yong Cui (cuiyong@tsinghua.edu.cn) is the corresponding author.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 26,2025 at 04:30:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3510936

instance, Huang et al. [9] proposed a domain-adapted LLM framework that integrates external network tools, while Wu et al. [10] introduced the use of multiple task-specific encoders to map multimodal network data into the token space of an LLM, allowing the LLM to function as a unified foundation model. Compared to these works, our comprehensive survey presents a condensed overview of existing LLMN research and the lessons learned from it. Moreover, based on the understanding of existing research, we have abstracted a universal workflow that includes essential mechanisms such as validation and feedback. This survey aims to provide a foundational and practical research roadmap for newcomers to the LLMN field, while also enabling experienced network professionals to quickly grasp the transformative potential of LLMs within their domain. Additionally, we hope this article serves as a catalyst for further research endeavors by inspiring networking and artificial intelligence (AI) experts to delve into more profound investigations.

II. Background

In this section, we first introduce the motivation and methodology of our literature survey. We then compare the differences between large language models (LLMs) and traditional methods, highlighting their respective strengths and limitations in the networking domain.

A. Motivation and Methodology of the Literature Survey

As LLMs have achieved success across various domains, network researchers are increasingly exploring their use in accomplishing complex networking tasks. At the 22nd ACM Workshop on Hot Topics in Networks (HotNets 2023), this interest was evident, with two of the nine sessions specifically dedicated to this topic. This underscores the growing intersection of LLMs and networking, sparking significant academic discussion.

To examine this trend, we conducted a systematic search of recent conference proceedings, journal articles, and arXiv preprints within the networking domain, using "large language models" and "networking" as keywords. Our analysis revealed a growing number of studies that adopt LLMs as a central methodology. To illustrate the impact of LLMs on various networking tasks, we focused on four key vertical fields, as outlined in reference [9]: network design, network configuration, network diagnosis, and network security. We first categorized the papers we reviewed into these four fields and then selected two pertinent papers from each for detailed discussion in Section IV.

B. Comprasion of LLMs and Traditional Approaches

In the development of solutions for networking tasks, researchers have explored various approaches. Early studies primarily focused on mathematical modeling, which often depended on a series of assumptions, resulting in insufficient accuracy. Additionally, the complex solving processes of this approach made it challenging to obtain precise answers. In order to achieve results within a limited time, approximate solving methods are often employed. The errors generated during the modeling and solving processes, which are difficult to eliminate, limit the scope of the application.

As machine learning technology has advanced, researchers have increasingly adopted it as a primary approach. Machine learning relies on large volumes of domain-specific data for training, enabling the development of high-performance models. These models can learn to make decisions for various tasks, such as prediction, classification, and generation. However, this approach is heavily dependent on domain-specific data and often lacks sufficient generalization capabilities. Moreover, machine learning methods face challenges when integrated with different tools, which limits their effectiveness in automating complex network tasks.

Recently, the emergence of LLMs has opened up new opportunities in networking. Trained on extensive datasets and equipped with large-scale parameters, these models offer several significant advantages. First, they possess logical reasoning abilities, enabling the efficient handling of complex tasks by leveraging domain knowledge. Second, they acquire diverse knowledge through large-scale pre-training. This approach reduces reliance on domain-specific data and enhances their generalization capabilities, allowing them to adapt to a wide range of tasks and environments. Furthermore, their ability to process natural language inputs and outputs facilitates the integration of network tools, thereby enhancing overall network intelligence. However, LLMs also face some challenges, such as the issue of hallucination and the lack of domain-specific knowledge. Therefore, when applying LLMs in the networking domain, researchers and practitioners must address these issues.

III. Basic Workflow for LLMN

In this section, we present an abstract and unified workflow for integrating LLMs into the networking domain. The proposed workflow, illustrated in Figure 1, consists of six key stages. While these stages are interconnected, they are not fully coupled, allowing for flexibility and adaptability in the implementation process. Each stage encompasses a range of unique techniques and considerations, which we will explore in depth.

A. Task Definition

It is important to present intricate networking tasks in a way that LLMs can understand. One approach is to assign personas to LLMs and provide initial global instructions that clarify the task's goals and requirements [5], [11]. While LLMs can capture contextual information to some extent, their short-term memory is limited. As a result, they may not perform well when dealing with longer text sequences or tasks that require long-term planning. To overcome this limitation, decomposing complex tasks into sub-tasks can be helpful. This allows for detailed inputs and expected outputs for each subtask. For example, the Incident Management (IM) process is time-consuming and labor-intensive due to the possible root causes being varied. Instead of directly incorporating LLMs to solve the end-to-end problem, the IM process can be broken

Authorized licensed use limited to: Tsinghua University. Downloaded on February 26,2025 at 04:30:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3510936

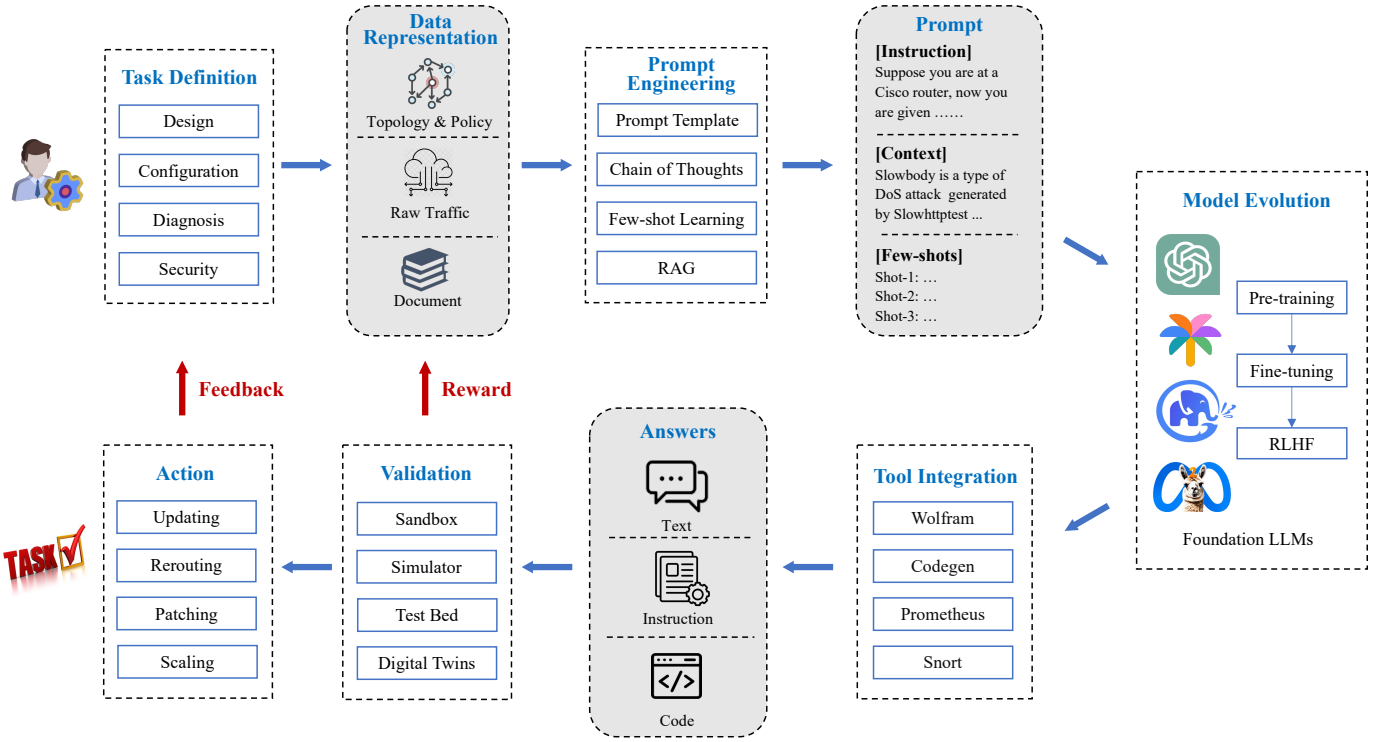

Fig. 1. The typical workflow of large language models for networking. RAG represents retrieval augmented generation, and RLHF represents reinforcement learning with human feedback.

down into three sub-tasks: hypothesis formation, hypothesis testing, and mitigation planning, where each of which can be completed independently by an LLM agent [11].

B. Data Representation

The objective of this stage is to handle heterogeneous data in the networking domain, including graph-based network topology, control plane policies, binary traffic data from the data plane, and text-based domain knowledge. Translating domain-specific information into natural language is an intuitive approach, e.g., describing the characteristics of traffic using natural language [8], but it may lack generality and lead to information loss. An alternative approach involves representing heterogeneous data by extracting features using various encoders [10], [12]. These features are then aligned with the token space of LLMs through methods such as linear projection. Additionally, for domain knowledge related to network tasks, a suitable approach is to store the preprocessed information in vector databases (e.g., Pinecone[1], Milvus[2]), which might enable efficient similarity searches and allow for effective retrieval of relevant knowledge. For example, to facilitate retrieving and analyzing operator data, comprehensive textual descriptions of various metrics and custom functions can be stored in vector databases for semantic search [6].

C. Prompt Engineering

Crafting effective prompts is crucial for guiding LLMs to produce desired outputs in networking tasks. Prompts should not only provide a detailed task description but also include problem-solving steps to support LLMs in making correct inferences. Constructing fine-grained chains of thought (CoT) is a common practice. Since LLMs trained on general corpora may have limited expertise in the networking domain, supplementing them with necessary knowledge through retrieval augmented generation (RAG) is crucial for ensuring the accuracy of responses. Additionally, the outputs of LLMs often serve as inputs for other tools or instructions for network device operations, requiring adherence to specific requirements. Techniques such as few-shot learning have proven effective in helping LLMs generalize and generate accurate results with minimal input-output examples. For example, to generate machine-readable grammars for different message types in the RTSP protocols, the prompt context includes the PLAY client request grammar for RTSP and the GET client request grammar for HTTP [7].

D. Model Evolution

Foundation LLMs play a crucial role in the LLMN workflow, and their capabilities significantly impact task completion. However, these models, pre-trained on general data, have limitations in addressing networking domain tasks. Enhancing LLM performance can be achieved by incorporating additional network-specific data or knowledge during various stages of training. Considering the high cost of training from scratch and the dynamic nature of network environments, fine-tuning LLMs using the latest network data is a common practice. This approach enables adaptation to network dynamics, providing relevant and effective solutions to current network challenges.

[1]Pinecone: https://www.pinecone.io/
[2]Milvus: https://milvus.io/

Authorized licensed use limited to: Tsinghua University. Downloaded on February 26,2025 at 04:30:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3510936

For example, in applying open-source LLMs to network data analysis for the 5G network, researchers utilized publicly available real 5G datasets to conduct instruction fine-tuning on the LLaMA 2 model, leading to the development of Mobile-LLaMA [13]. Moreover, techniques such as reinforcement learning with human feedback (RLHF) allow models to learn from the expertise of network professionals, generating outputs that align better with the context and values of the networking domain.

E. Tool Integration

The integration of various tools empowers LLMs to address a diverse range of complex problems. Considering LLMs may struggle with mathematical deduction, enhancing the accuracy of numerical results is crucial for decision-making in networking. One approach is to leverage mathematical tools like Wolfram[3] or utilize Python code for data analysis and calculation [2], [6]. In addition to general tools, detailed descriptions and usages of domain-specific tools can be provided to enable LLMs to automatically determine and select the appropriate tool based on different task requirements. For example, when it comes to synthesizing router configurations, LLMs alone are not effective. However, by incorporating existing configuration verifiers like Batfish[4] and Campion[5], which offer actionable feedback to address uncertainties and invalidities, LLMs can significantly reduce the need for human intervention [3].

F. Validation

Validation is an indispensable stage in the LLMN workflow to assess the correctness and security of LLM-generated outputs, which can include textual analysis reports, graph manipulation code, and network device instructions. To minimize syntax and semantic errors in LLM-generated code, techniques like self-consistency and self-debugging can be employed [2]. However, code that is syntactically and semantically correct may still lack proper functionality. Simulation and emulation are commonly used techniques that can help verify whether the code and operations generated by LLM have achieved the expected functionality. Furthermore, before deploying LLM-generated code and operations in a production environment, thorough risk monitoring and control measures should be implemented. An execution sandbox, for instance, creates an isolated environment to run uncertain or potentially harmful code, mitigating any adverse effects on the actual network [2].

The key stages mentioned above collectively determine the efficiency and performance of leveraging LLMs in handling various network tasks. Task definition focuses on describing and decomposing complex tasks, while data representation handles different data types. Prompt engineering guides LLMs in completing tasks, and model evolution concentrates on the model's performance. Tool integration extends LLM's capabilities, and validation is used to assess the correctness and security of the output. Furthermore, we underscore that iterative optimization can occur within each of these stages through two distinct feedback mechanisms: an internal loop driven by validation outcomes and an external loop guided by task execution results. Validation outcomes can steer the continuous learning and enhancement of LLMs, leading to the generation of outputs that are more precise and high-performing. Task execution feedback, on the other hand, can assist network administrators in refining execution procedures tailored to various network task scenarios, thereby promoting smoother integration of LLMs.

[3]Wolfram: https://www.wolframalpha.com/

[4]Batfish: https://batfish.org/

[5]Campion: https://github.com/atang42/batfish/tree/rm-localize

IV. Overview of Recent Advances

In this section, we provide a targeted overview of the up-to-date advances in applying LLMs to the networking domain. The discussion primarily focuses on the use of LLMs in four vertical fields: network design, network configuration, network diagnosis, and network security. However, the potential applications of LLMs extend far beyond these areas. Our aim is to highlight the ways in which LLMs are utilized, the transformative changes they can bring, and the potential limitations they may face. Table I outlines the approaches taken by these studies at every stage of the LLMN workflow.

A. Network Design

Network design involves detailed planning processes like protocol selection, bandwidth allocation, and network topology optimization. The challenge lies in making optimal decisions amidst ever-evolving technology, complex requirements, and limited resources. Traditionally, network engineers have relied on professional expertise, manual adjustments, and basic tools for design and optimization. LLMs, however, offer a promising alternative by combining vast networking knowledge with advanced reasoning and generation capabilities. They enhance network design by proposing tailored strategies for specific goals and enabling automated evaluation of design options.

As a part of initial investigations, several pieces of literature focus on the critical importance of data transmission and topology management in network architecture. He et al. [1] proposed leveraging LLM to develop adaptive bitrate streaming (ABR) algorithms, a pivotal component for dynamically adjusting video quality in streaming media delivery. Given the challenge of directly crafting high-quality algorithms with LLM, they incorporated the classic Pensieve algorithm as part of the input prompts. Their objective was to ask LLM to enhance the original algorithm and generate a range of candidate algorithms. As not all these algorithms exhibited significant enhancements, evaluating these algorithms in a network simulator was imperative to identify the most effective one. Experimental findings demonstrate LLM's capability to produce outstanding algorithms tailored to diverse network conditions. Mani et al. [2] have investigated the use of LLMs for code generation in graph analysis and manipulation for topology management. They proposed a structured framework consisting of application wrappers, application prompt generators, code-generation prompt generators, and execution sandboxes. This framework generates user query-based

Authorized licensed use limited to: Tsinghua University. Downloaded on February 26,2025 at 04:30:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3510936

TABLE I
RELATIONSHIPS BETWEEN THE LATEST ADVANCES AND KEY STAGES IN LLMN WORKFLOW.

Research Work	Vertical Field	Task Definition	Data Representation	Prompt Engineering	Model Evolution*	Tools Integration	Validation
He et al. [1]	Design	Designing adaptive bitrate streaming (ABR) algorithms	Representing the original design of states and network architectures with Python functions	CoT & Prompt Template	Off-the-shelf GPT-3.5 & GPT-4	/	Evaluation of various candidate algorithms in a network simulator
Mani et al. [2]		Generating code for graph analysis and manipulation	Representing network topology with different formats, e.g. JSON, SQL, Pandas, NetworkX	CoT & Prompt Template & RAG	Off-the-shelf LLMs like GPT-4, GPT-3, text-davinci-003 and Google Bard	NetworkX with flexible API	Execution in a sandbox environment
Mondal et al. [3]	Configuration	Synthesizing router configurations	Representing network topology and connection with JSON	One shot learning & Prompt Template	Off-the-shelf GPT-4	Syntax verifier & Semantics verifier	Manual correction
Lian et al. [4]		Configuration validation to detect errors	Storing labeled valid configurations and misconfiguration in vector databases	Prompt template, Few-shot learning & RAG	Off-the-shelf LLMs like Code-Davinci-002, GPT-3.5-turbo, GPT-4 and Babbage-002	/	Aggregation of outputs from various LLMs by voting
Kotaru [6]	Diagnosis	Retrieval and analysis of operator data	Stroring definitions of various metrics and custom functions in vector databases	Few-shot learning & RAG	Off-the-shelf LLMs like GPT-4, GPT3.5-turbo and text-curie-001	/	Execution in a sandbox environment & Creating GitHub issues to attain feedback from experts
Zhou et al. [5]		Assisting in conducting network incident investigations as a research assistant	Storing the most relevant domain knowledge in a JSON file	CoT & RAG	Off-the-shelf GPT-4	Search engines	Self-test & Self-learning
Meng et at. [7]	Security	Assisting in protocol fuzzing	Representing protocol formats with specific strings patterns	Few-shot learning	Off-the-shelf GPT-3.5-turbo	/	Self-consistency check
Wang et al. [8]		Providing in-depth analysis and customized mitigation recommendations for DDoS attack	Representing raw traffic with LLM-understandable texts	Prompt template	Off-the-shelf GPT-4	/	/

* Most of the existing research works have only utilized off-the-shelf LLMs without specifically evolving these models, which we believe should be the focus of future research.

prompts for specific network applications, guides LLMs in producing practical code, and provides a secure execution environment for this code. Their findings reveal that this method significantly improves code quality over direct LLM processing of raw network data, although code quality tends to decline with increasing task complexity. The study suggests that future efforts should aim to enhance LLMs' capabilities in managing complex network tasks.

B. Network Configuration

Network configuration plays a critical role in ensuring the proper functioning of network devices and applications. However, this task is complicated by the multitude of settings and parameters, which require a deep understanding of network concepts and operational details. Moreover, the dynamic nature of network conditions and business objectives requires configurations to be regularly updated to align with emerging needs. Traditionally, network configurations have been manually crafted and verified, a process that is not only time-consuming but also prone to errors. In contrast, LLMs excel in identifying associations within complex contexts, enabling them to manage intricate configuration files and comprehend the relationships among various configuration items. Additionally, LLMs leverage few-shot learning and RAG capabilities, which significantly enhance their effectiveness in generating network configurations.

Inspired by the advanced capabilities of LLMs in aiding with programming tasks, Mondal et al. [3] have explored

Authorized licensed use limited to: Tsinghua University. Downloaded on February 26,2025 at 04:30:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3510936

the application of LLMs, such as GPT-4, in synthesizing network router configurations. They introduced a novel approach, *Verified Prompt Programming*, which combines LLMs with a verification system to automatically correct errors using localized feedback, thus enhancing the accuracy of the generated configurations. Their research focused on practical applications, such as converting router configurations from Cisco to Juniper format and implementing a no-relay policy across routers. Despite the advancements, they emphasize the continued importance of human oversight in refining LLM-generated configurations to ensure the precision required for router setups. Aimed at enhancing the process of configuration validation prior to deployment, Lian et al. [4] have developed a novel LLM-based framework named Ciri. This approach marks a significant shift from traditional methods that rely heavily on manual checks or extensive engineering towards a more efficient and automated solution. Ciri leverages advanced prompt engineering with few-shot learning, drawing on existing configuration data to identify and explain potential misconfigurations from either complete files or diffs. Furthermore, Ciri enhances reliability by aggregating insights from multiple LLMs, such as Code-Davinci-002, GPT-3.5-turbo, and GPT-4, through a voting mechanism. This strategy effectively mitigates issues like hallucination and inconsistency, which are common in LLM outputs. However, while Ciri excels at detecting syntax and range errors, it encounters challenges with configurations that involve intricate dependencies or specific software version requirements. Overcoming these limitations may involve retraining or fine-tuning LLMs with targeted data on dependencies and versioning.

C. Network Diagnosis

Network diagnosis is critical for addressing network issues, encompassing data collection (like system logs and traffic data), data analysis (to spot abnormal patterns or faults), identifying root causes, and implementing fixes. Given the complexity of network environments and the variety of potential issues, network diagnosis presents significant challenges. Typically, administrators might sift through extensive monitoring data and consider numerous potential causes, a process that demands a deep understanding of diverse systems. In contrast, LLMs present a more advanced solution by directly providing administrators with key data and relevant metrics, as well as proposing potential solutions.This approach not only increases the efficiency of fault diagnosis but also enhances accuracy, enabling more intelligent and automated network diagnosis.

In commercial network operations, managing thousands of counters and metrics for data analysis is a complex task. To reduce reliance on specialists and expedite issue resolution, Kotaru [6] introduced the Data Intelligence for Operators Copilot (DIO Copilot), a natural language interface utilizing LLMs for efficient data retrieval and analysis. DIO Copilot enhances metric extraction through semantic search and employs few-shot learning, enabling LLMs to accurately respond to user queries with minimal examples. Notably, it features a mechanism for generating GitHub issues to foster expert collaboration and knowledge updates. When evaluated against leading natural language interfaces for databases, DIO Copilot demonstrated superior performance, achieving 66% execution accuracy on a benchmark dataset. Besides data retrieval and analysis, recent studies [5], [11] have explored the use of LLMs in incident management and investigation. Hamadanian et al. [11] have proposed a holistic theoretical framework aimed at building an AI helper for incident management. This framework is guided by three fundamental principles: iterative prediction, reliable & safe, and adaptive. In contrast to the one-shot feed-forward model, iterative prediction can successfully handle more complex (often more impacting) or novel incidents by hypothesizing, testing, and re-evaluating its decisions in a feedback loop. Concurrently, Zhou et al. [5] developed an interactive software agent aimed at aiding in the investigation of internet incidents. This agent is structured around four key components: role definition, information retrieval, knowledge memory, and knowledge testing with self-learning capabilities. Utilizing AutoGPT[6], the agent autonomously gathers relevant information from the internet, enhancing its reasoning and query response abilities. It continuously enriches its knowledge base until it achieves sufficient confidence to function akin to a researcher. In a practical test, an agent named Bob was trained to autonomously learn about solar storm events, successfully providing expert-level insights into their effects on the internet and demonstrating capabilities comparable to professional researchers.

D. Network Security

Network security is a continuous cycle of offense and defense involving multiple stages, such as threat modeling, risk assessment, attack identification, and incident response. In this process, defenders implement various strategies to strengthen systems while attackers continuously devise more complex techniques to breach defenses. The emergence of LLMs provides robust support for ensuring security due to their comprehensive understanding of network security and ability to leverage external tools. They not only assist in detecting potential vulnerabilities but also generate customized defense strategies tailored to specific threats, thereby enhancing the efficiency of security teams in mitigating network attacks.

Mutation-based protocol fuzzing is a security testing technique that assesses protocol implementations by fuzzing seed messages, aiming to uncover potential vulnerabilities. However, the limited diversity of seed messages poses a constraint. To address this limitation, Meng et al. [7] proposed ChatAFL, a novel engine that enhances existing protocol fuzzing tools using LLMs. ChatAFL builds upon AFLNet and introduces machine-readable protocol grammar extraction to enrich seed inputs' diversity. Moreover, when the fuzzer exhausts exploration of new states, ChatAFL interacts with the LLM to generate client requests that induce server transitions. Experimental results demonstrate that compared to the baseline AFLNet, ChatAFL achieves improved state transition coverage, state coverage, and code coverage, while also identifying more security vulnerabilities. Considering existing AI-driven methods for DDoS defense have certain limitations

[6]AutoGPT: https://github.com/Significant-Gravitas/AutoGPT

Authorized licensed use limited to: Tsinghua University. Downloaded on February 26,2025 at 04:30:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3510936

in practical application, including lack of explainability and lack of mitigation instructions, Wang et al. [8] have proposed a LLM-based DDoS defense framework named ShieldGPT to solve these shortcomings. ShieldGPT comprises four modules: attack detection, traffic representation, domain-knowledge injection, and role representation. It transforms raw traffic data into LLM-friendly text, maintaining essential features. By integrating this data with domain knowledge using prompt templates, ShieldGPT enables the LLM to elucidate attack behaviors and propose specific mitigation actions. Preliminary experiments demonstrate that ShieldGP can provide in-depth attack analysis and customized mitigation recommendations, laying the foundation for establishing a truly autonomous DDoS defense system.

V. Challenges and Future Prospects

Previous research carried out across various fields aligns well with the proposed LLMN workflow, indicating its universal applicability and providing guidance for network researchers to advance their studies. However, upon analyzing these existing works, we have identified several critical challenges at each stage of the workflow that require further research efforts. This section outlines the key challenges and potential opportunities as follows.

A. Intelligent Planning

Intelligent planning capabilities are crucial for accomplishing complex tasks that involve long-term objectives and multi-step decision-making. In the networking domain, many of these tasks still heavily rely on human labor and expert knowledge. A promising direction is the development of LLM-based agents that integrate perception, decision-making, and action capabilities [14]. This integration enables these agents to assess situations more accurately and generate more effective action strategies based on feedback, facilitating rapid iteration.

B. Understanding Network Traffic

Given the critical role of traffic data in the networking domain, enabling LLMs to comprehend network traffic has emerged as a significant research direction. One potential solution is to implement a traffic token generator to extend the token space of LLMs, thereby enhancing their understanding and reasoning capabilities through fine-tuning. Another approach involves introducing extensive traffic data during the pre-training phase and designing multi-objective pre-training tasks.

C. Prompt Engineering for Deliberate Reasoning

Many networking tasks involve integrating multiple intermediate results to reach a final conclusion. Recent research has made progress in addressing these multi-path reasoning processes by employing various methods. Notably, approaches such as the Tree of Thoughts(ToT[7])and Graph of Thoughts (GoT[8]), that mimic human thinking and brain mechanisms, significantly enhance LLMs' problem-solving abilities. Furthermore, designing effective reflective prompts that allow the model to learn from failures and generate more accurate results is also a worthwhile area of research.

D. Network-specific LLMs

The construction of LLMs specialized for the networking domain is a promising approach to enhancing efficiency and performance. On the one hand, it is essential to utilize diverse network data from various scenarios during the pre-training phase. Based on such a foundational model, continuous fine-tuning should then be applied to adapt the model for specific targeted tasks [15]. On the other hand, considering the temporal and spatial constraints present in network environments, it is crucial that these models remain lightweight and efficient. Notably, technologies such as knowledge distillation, model quantization, and novel frameworks warrant further attention.

E. Autonomous Tools Utilization

The reliance on few-shot learning often results in incomplete tool comprehension and suboptimal performance in LLMs. Additionally, fine-tuning LLMs with tool-specific documentation creates high overhead and reduces flexibility. A potential solution is to represent tools as tokens, allowing LLMs to learn their embedded representations and generate tool tokens as needed. This approach enables the model to alternate between reasoning mode and tool mode, maximizing its ability to handle both natural language and tool-specific tasks.

F. Validation Environment

Ensuring the reliability and safety of applying LLMs in networking presents a critical challenge. We propose that integrating LLMs with validation environments, such as digital twins, can help control and mitigate the risks associated with real-world networks. These validation environments simulate system behavior, operate synchronously with actual systems, monitor system status, and provide real-time feedback. This feedback enables continuous adjustments and optimization of the LLMs' outputs, leading to an effective iteration.

The aforementioned challenges and potential future directions seek to address issues that may arise at various stages of the workflow. Additionally, we explore trends that have not been sufficiently studied from a broader perspective. For instance, future research may focus on enhancing network architecture to meet the requirements of low latency and high quality of service (QoS) when applying LLMs in networking. In this context, innovative technologies such as edge computing and 5G/6G will play a crucial role. These technologies can provide computing resources closer to the data source. By efficiently utilizing edge resources, they can accelerate the fine-tuning and inference of LLMs in resource-constrained scenarios, thereby achieving more real-time data processing and task execution.

[7]ToT: https://github.com/princeton-nlp/tree-of-thought-llm

[8]GoT: https://github.com/spcl/graph-of-thoughts

Authorized licensed use limited to: Tsinghua University. Downloaded on February 26,2025 at 04:30:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3510936

VI. Conclusion

In this paper, we provide a foundational workflow as a practical guide for researchers to explore novel applications of LLMs in networking research. For a deeper comprehension, we summarize the recent advances of LLMs in the networking domain, covering various important subfields, including design, configuration, diagnostics, and security. Furthermore, there are still many unresolved challenges and we shed light on directions for further research efforts aimed at better integrating LLMs with networking.

Acknowledgments

This work was supported by the NSFC project under Grant 62132009 and Grant 62221003.

References

[1] Z. He, A. Gottipati, L. Qiu, F. Y. Yan, X. Luo, K. Xu, and Y. Yang, "Llm-abr: Designing adaptive bitrate algorithms via large language models," *arXiv preprint arXiv:2404.01617*, 2024.

[2] S. K. Mani, Y. Zhou, K. Hsieh, S. Segarra, T. Eberl, E. Azulai, I. Frizler, R. Chandra, and S. Kandula, "Enhancing network management using code generated by large language models," in *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, 2023, pp. 196–204.

[3] R. Mondal, A. Tang, R. Beckett, T. Millstein, and G. Varghese, "What do llms need to synthesize correct router configurations?" in *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, 2023, pp. 189–195.

[4] X. Lian, Y. Chen, R. Cheng, J. Huang, P. Thakkar, and T. Xu, "Configuration validation with large language models," *arXiv preprint arXiv:2310.09690*, 2023.

[5] Y. Zhou, N. Yu, and Z. Liu, "Towards interactive research agents for internet incident investigation," in *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, 2023, pp. 33–40.

[6] M. Kotaru, "Adapting foundation models for operator data analytics," in *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, 2023, pp. 172–179.

[7] R. Meng, M. Mirchev, M. Böhme, and A. Roychoudhury, "Large language model guided protocol fuzzing," in *Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS)*, 2024.

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

[9] Y. Huang, H. Du, X. Zhang, D. Niyato, J. Kang, Z. Xiong, S. Wang, and T. Huang, "Large language models for networking: Applications, enabling techniques, and challenges," *IEEE Network*, 2024.

[10] D. Wu, X. Wang, Y. Qiao, Z. Wang, J. Jiang, S. Cui, and F. Wang, "Netllm: Adapting large language models for networking," in *Proceedings of the ACM SIGCOMM 2024 Conference*, 2024, pp. 661–678.

[11] P. Hamadanian, B. Arzani, S. Fouladi, S. K. R. Kakarla, R. Fonseca, D. Billor, A. Cheema, E. Nkposong, and R. Chandra, "A holistic view of ai-driven network incident management," in *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, 2023, pp. 180–188.

[12] X. Xiao, W. Xiao, R. Li, X. Luo, H. Zheng, and S. Xia, "Ebsnn: Extended byte segment neural network for network traffic classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 5, pp. 3521–3538, 2021.

[13] K. B. Kan, H. Mun, G. Cao, and Y. Lee, "Mobile-llama: Instruction fine-tuning open-source llm for network analysis in 5g networks," *IEEE Network*, 2024.

[14] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, 2023.

[15] F. Le, M. Srivatsa, R. Ganti, and V. Sekar, "Rethinking data-driven networking with foundation models: challenges and opportunities," in *Proceedings of the 21st ACM Workshop on Hot Topics in Networks*, 2022, pp. 188–197.

Chang Liu is currently pursuing a Ph.D. degree with the Department of Computer Science and Technology at Tsinghua University, Beijing, China. He received the B.E. and M.E. degrees in the School of Computer Science, National University of Defense Technology, Changsha, China, in 2013 and 2015, respectively. His current research interests include network security and artificial intelligence.

Xiaohui Xie received the B.E. and Ph.D. degrees in computer science and engineering from Tsinghua University, China, in 2016 and 2021, respectively. He is currently an Assistant Professor at the Computer Science Department, Tsinghua University. His research interests include Network Anomaly Detection, Information Retrieval, and Artificial intelligence.

Xinggong Zhang (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science, Peking University, Beijing, China, in 2011. He is currently an Associate Professor with the Wangxuan Institute of Computer Technology, Peking University. Before that, he was a Senior Researcher with the Founder Research and Development Center at Peking University from 1998 to 2007. He was a visiting scholar at the Polytechnic Institute of New York University from 2010 to 2011. His research interests include the modeling and optimization of multimedia networks, VR/AR/video streaming, and satellite networks.

Yong Cui (Member, IEEE) received the B.E. and Ph.D. degrees in computer science and engineering from Tsinghua University, China, in 1999 and 2004, respectively. He is currently a Full Professor at the Computer Science Department, Tsinghua University. His major research interests include mobile cloud computing and network architecture. He served or serves on the editorial boards of the IEEE TPDS, IEEE TCC, IEEE Internet Computing, and the IEEE Network.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 26,2025 at 04:30:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.