future internet

MDPI

Review

Large Language Models Meet Next-Generation Networking Technologies: A Review

Ching-Nam Hang [1], Pei-Duo Yu [2], Roberto Morabito [3] and Chee-Wei Tan [4,*]

[1] Yam Pak Charitable Foundation School of Computing and Information Sciences, Saint Francis University, Hong Kong, China; cnhang@sfu.edu.hk
[2] Department of Applied Mathematics, Chung Yuan Christian University, Taoyuan City 320314, Taiwan; peiduoyu@cycu.edu.tw
[3] Communication Systems Department, EURECOM, 06140 Biot, France; roberto.morabito@eurecom.fr
[4] College of Computing and Data Science, Nanyang Technological University, Singapore 639798, Singapore
* Correspondence: cheewei.tan@ntu.edu.sg

Abstract: The evolution of network technologies has significantly transformed global communication, information sharing, and connectivity. Traditional networks, relying on static configurations and manual interventions, face substantial challenges such as complex management, inefficiency, and susceptibility to human error. The rise of artificial intelligence (AI) has begun to address these issues by automating tasks like network configuration, traffic optimization, and security enhancements. Despite their potential, integrating AI models in network engineering encounters practical obstacles including complex configurations, heterogeneous infrastructure, unstructured data, and dynamic environments. Generative AI, particularly large language models (LLMs), represents a promising advancement in AI, with capabilities extending to natural language processing tasks like translation, summarization, and sentiment analysis. This paper aims to provide a comprehensive review exploring the transformative role of LLMs in modern network engineering. In particular, it addresses gaps in the existing literature by focusing on LLM applications in network design and planning, implementation, analytics, and management. It also discusses current research efforts, challenges, and future opportunities, aiming to provide a comprehensive guide for networking professionals and researchers. The main goal is to facilitate the adoption and advancement of AI and LLMs in networking, promoting more efficient, resilient, and intelligent network systems.

Keywords: large language models; generative AI; network intelligence; next-generation network; networked AI systems; edge intelligence; networks

Citation: Hang, C.-N.; Yu, P.-D.; Morabito, R.; Tan, C.-W. Large Language Models Meet Next-Generation Networking Technologies: A Review. *Future Internet* **2024**, *16*, 365. https://doi.org/10.3390/fi16100365

Academic Editor: Gianluigi Ferrari

Received: 5 August 2024
Revised: 6 September 2024
Accepted: 24 September 2024
Published: 7 October 2024

1. Introduction

The evolution of network technologies has dramatically transformed the way we communicate, share information, and connect with the world [1–3]. Traditional networks, which rely heavily on static configurations and manual interventions, are foundational to modern infrastructure but come with significant challenges. These networks require complex management and operational expertise, making them difficult for non-experts to handle. Meanwhile, as networks increasingly grow in size and complexity, managing them becomes even more daunting. Issues such as network configuration, optimization, troubleshooting, and security in large-scale network environments managed through labor-intensive processes can lead to inefficiencies and human errors [4]. As the demand for more agile, resilient, and secure networks grows, these traditional methods fall short, highlighting the need for more intelligent and automated solutions.

Over the past decade, artificial intelligence (AI) has profoundly reshaped the landscape of computer technologies, driving advancements across numerous sectors [5,6]. Building on this momentum, AI has emerged as a transformative force in addressing the limitations of traditional network management. AI algorithms can deliver high-quality solutions

such as automating network configuration, optimizing traffic flows, enhancing security protocols, and predicting network failures before they occur, thereby significantly reducing the need for manual interventions and minimizing human errors [3,7]. For instance, machine learning techniques analyze vast amounts of network data to identify patterns and anomalies, enabling predictive maintenance that can foresee hardware failures and trigger preventive actions, thus minimizing downtime and maintaining network reliability [8,9]. The application of AI in networking also enhances network security by automatically detecting and responding to threats in real time, which is especially critical in large-scale, dynamic environments [1]. To enhance AI algorithms, mathematical optimization theory is a common approach [10–13]. However, AI-driven solutions are often inherently complex and unexplainable for human understanding. This lack of interpretability significantly limits the commercial adoption of AI-based solutions in practice [14].

To address these challenges, generative AI (GenAI), the next frontier in AI, can play a pivotal role in transforming networks. GenAI encompasses generative models capable of creating content and understanding complex patterns across various domains. These models can generate text, images, videos, music, and code, demonstrating a sophisticated grasp of the data they process. Consequently, the applications of GenAI are wide-ranging, from aiding creative industries and automating routine tasks to advancing scientific research through simulation and hypothesis generation. Recently, large language models (LLMs) have drawn considerable attention from the research community as a groundbreaking technology in the field of GenAI. LLMs excel in natural language processing (NLP) tasks, such as language translation, text summarization, and sentiment analysis. These models can interpret and generate human-like text based on extensive datasets, making them powerful tools for any field that involves intricate linguistic tasks. To further enhance the capabilities of LLMs, techniques such as prompt engineering [15–18], fine-tuning [19,20], and retrieval-augmented generation (RAG) [21,22] can be employed to develop domain-specific LLMs [23–27]. These methods enable LLMs to effectively solve specific tasks in various domains, including network engineering.

LLMs are currently a highly popular research area, with the technology still in its early stages of development, offering numerous potential avenues for exploration and innovation. Meanwhile, network engineering remains a critical and long-established field facing numerous complex challenges. Given the growing trend of using natural language to address networking issues, exploring the convergence of the emerging fields of LLMs and network engineering presents significant potential. Therefore, conducting a comprehensive exploration to understand how LLMs can revolutionize network management and operations is essential. While most existing reviews focus broadly on traditional AI applications in networks, there is a noticeable gap in the literature specifically focusing on the role of LLMs in this domain. Although some recent reviews discuss LLMs and networks, they focus solely on a specific network area, which may not be comprehensive. The rapid evolution of LLMs, with new models continually emerging and offering more advanced features and capabilities, underscores the pressing need for a comprehensive review that covers more critical areas of network topics. More importantly, such a review can highlight current advancements and serve to provide insights into the most recent and sophisticated developments in LLMs for networking. Table 1 summarizes the highly cited or recently published review papers on AI for networks from 2018 to 2024, marking the period since the introduction of the first Generative Pre-trained Transformer (GPT) in 2018 [28].

The purpose of this review paper is to address the identified gap in two stages. First, we explore the applications of AI and LLMs in modern networking and discuss current research efforts to interpret AI and LLMs for networks. Second, we outline the open challenges and prospects in this area. In essence, our goal is to provide a practical guide on AI and LLMs for professionals and researchers in the networking community and to promote the continued advancement of AI and LLMs in networking.

Table 1. A comparison of related reviews on AI in networking.

Reference	Year	Technological Aspect		Application Field
		AI	LLMs	
Mata et al. [29]	2018	✓	✗	Optical Networks
Kibria et al. [30]	2018	✓	✗	Next-Generation Wireless Networks
Xie et al. [31]	2018	✓	✗	Software-Defined Networking
Cayamcela et al. [32]	2018	✓	✗	5G Mobile and Wireless Networks
Zhao et al. [33]	2019	✓	✗	Software-Defined Networking
Elsayed et al. [34]	2019	✓	✗	Future Wireless Networks
Chen et al. [35]	2019	✓	✗	Edge Computing
Zhang et al. [36]	2019	✓	✗	Mobile and Wireless Networking
Sun et al. [37]	2019	✓	✗	Wireless Networks
Wang et al. [38]	2020	✓	✗	5G Wireless Networks
Nguyen et al. [39]	2020	✓	✗	Wireless Networks
Semong et al. [40]	2020	✓	✗	Software-Defined Networking
Zeydan et al. [41]	2020	✓	✗	Intent-Based Networking
Mukhopadhyay et al. [42]	2021	✓	✗	Internet of Things
Chang et al. [43]	2021	✓	✗	Edge Computing
Letaief et al. [44]	2021	✓	✗	6G Networks
Murshed et al. [45]	2021	✓	✗	Edge Computing
Song et al. [46]	2022	✓	✗	Networking Systems
Gupta et al. [47]	2022	✓	✗	Mobile Networks
Macas et al. [48]	2022	✓	✗	Cybersecurity
Salau et al. [49]	2022	✓	✗	Wireless Networks
Singh et al. [50]	2023	✓	✗	Edge Computing
Zuo et al. [51]	2023	✓	✗	6G Networks
Bourechak et al. [52]	2023	✓	✗	Edge Computing
Gao [53]	2023	✓	✓	Cybersecurity
Tarkoma et al. [54]	2023	✓	✓	6G Systems
Gill et al. [55]	2024	✓	✗	Edge Computing
Alhammadi et al. [56]	2024	✓	✗	6G Wireless Networks
Ospina Cifuentes et al. [57]	2024	✓	✗	Software-Defined Networking
Chen et al. [58]	2024	✓	✗	6G Wireless Networks
Ozkan-Ozay et al. [59]	2024	✓	✓	Cybersecurity
Celik et al. [60]	2024	✓	✓	6G Networks
Khoramnejad et al. [61]	2024	✓	✓	Next-Generation Wireless Networks
Bhardwaj et al. [62]	2024	✓	✓	Edge Computing
Karapantelakis et al. [63]	2024	✓	✓	Mobile Networks
Zhou et al. [64]	2024	✓	✓	Telecommunications
This Study	2024	✓	✓	Next-Generation Network

To summarize, the contributions of this paper are as follows:

- We provide a comprehensive review of the potential applications of LLMs across the four key stages of network engineering: Network Design and Planning, Network Implementation, Network Analytics, and Network Management.
- We present strategies for enhancing network performance and management using network intelligence, highlighting the limitations of current AI-driven methods. Additionally, we demonstrate how LLMs can address these limitations and improve existing network intelligence.
- We identify key challenges in integrating LLMs into network engineering and, based on recent developments, provide future research directions to further optimize network functionality and efficiency through the use of LLMs and AI.

This paper is organized as follows. In Section 2, we provide a background on network intelligence and LLMs, detailing how LLMs can be applied to networking. In Section 3, we explore application domains where LLMs can be beneficial for networks. Section 4 discusses the current open research challenges and suggests future research opportunities related to LLMs in networking. We conclude the paper in Section 5.

2. Background

2.1. Network Intelligence

Traditional network development and infrastructure management present significant challenges due to their complexity and the intensive knowledge and labor required. These tasks demand substantial expertise and manual effort, as they involve continuous adaptation to evolving technical standards and protocols. The shift from 5G to 6G, for instance, necessitates updating numerous protocols, posing a significant burden on network engineers. Additionally, critical tasks such as network performance monitoring and fault diagnosis are formidable technical obstacles. In complex network environments, effectively monitoring performance, as well as detecting and locating faults promptly, is a substantial challenge, requiring sophisticated tools and strategies. To overcome these issues, network intelligence leverages advanced computational techniques to enhance the management and optimization of network systems. Network intelligence refers to the application of intelligent methodologies to automate and improve various aspects of network operations [65–67]. Common advanced technologies include deep learning [68,69], machine learning [70,71], transfer learning [72,73], and reinforcement learning [74,75]. Each of these models can be tailored to specific environments and tasks. For example, adaptive routing algorithms can improve quality of service (QoS), edge intelligence can optimize high-performance computing [76,77], automated configuration synthesis modules can reduce manual errors, and intelligent assistants can enhance customer service. These intelligence-driven approaches enable more efficient, responsive, and resilient network management, paving the way for smarter and more adaptable networking solutions.

In the realm of network engineering, there are primarily four stages: network design and planning, network implementation, network analytics, and network management. As illustrated in Figure 1, these stages encompass a comprehensive lifecycle of network systems, from initial design to ongoing management. Network intelligence offers transformative potential in addressing the limitations of traditional approaches across these stages in network engineering. We provide an overview below of how network intelligence can be incorporated to enhance network engineering:

- *Network Design and Planning*: Efficient network design and planning are essential for ensuring optimal network performance and resource utilization. This process involves developing the network architecture, such as capacity planning, resource scheduling, and load balancing. Typically, these tasks are labor-intensive, demanding considerable expertise and manual intervention, and they require ongoing adaptation to emerging standards and protocols. AI algorithms can enhance these tasks by automating the design process, increasing precision, and minimizing the dependence on manual expertise.

- *Network Implementation*: Successful network implementation is essential for the deployment of robust and efficient network infrastructures. This task involves the deployment, configuration, and documentation of network infrastructure. Manual implementation can be error-prone and time-consuming, requiring extensive coordination and detailed configuration. AI-driven tools can streamline deployment and configuration processes, ensuring precise execution and significantly reducing setup time and effort.
- *Network Analytics*: Continuous network analytics are vital for maintaining network health and performance. This process focuses on monitoring and analyzing network performance through traffic analysis, log analysis, and behavior analysis. Effective monitoring and fault diagnosis are challenging due to the complexity of modern networks, leading to potential delays in issue detection and resolution. Advanced AI models can continuously analyze network data, predict potential issues, and provide proactive maintenance insights, enhancing reliability and efficiency.
- *Network Management*: Efficient network management is critical for the ongoing operation and security of a network. This process involves ongoing monitoring, optimization, and security protection of the network. Manual management can be inefficient, with a high risk of human error, and it requires constant adaptation to changing network conditions. Intelligent systems can autonomously manage monitoring, optimization, and security tasks, adapting in real time to network changes and reducing the need for manual interventions. For example, AI-driven systems can enhance network security by automatically detecting and responding to threats, allowing for proactive threat mitigation. Additionally, AI models can continuously adapt security protocols by learning from past incidents, improving its ability to prevent and counter new, evolving threats, thus ensuring more robust and adaptive network protection.

AI methods are often viewed as a black box by application developers and network administrators. However, deploying AI models to address networking issues involves numerous practical challenges. We identify four main practical problems associated with implementing AI models in real-world network intelligence:

- *Complex Network Configurations*: AI models can help automate and simplify the configuration of complex networks, but accurately translating high-level management policies into network commands can still be a challenge.
- *Heterogeneous Infrastructure Management*: Network provisioning often involves intricate configurations and updates tightly coupled with underlying heterogeneous and diverse infrastructure, necessitating an abstraction layer for network operators to manage network parameters independently of the underlying infrastructure [41]. Traditional AI algorithms struggle to navigate and integrate with such diverse systems, making effective management challenging.
- *Unstructured Log and Network Data*: Networks generate vast amounts of unstructured log and operational data, making it difficult to extract actionable insights. These unstructured data complicate troubleshooting, diagnostics, and overall network management, presenting a significant challenge for traditional AI techniques that rely on clear, structured inputs to function effectively.
- *Dynamic Network Environment*: Conventional AI systems often rely on static, rule-based learning models, which do not adapt well to dynamic network environments. These systems struggle to continuously interpret and adjust to changes in network intents, such as user complaints or system alerts. This limitation results in less accurate and effective responses, preventing the network from staying aligned with evolving objectives and conditions.

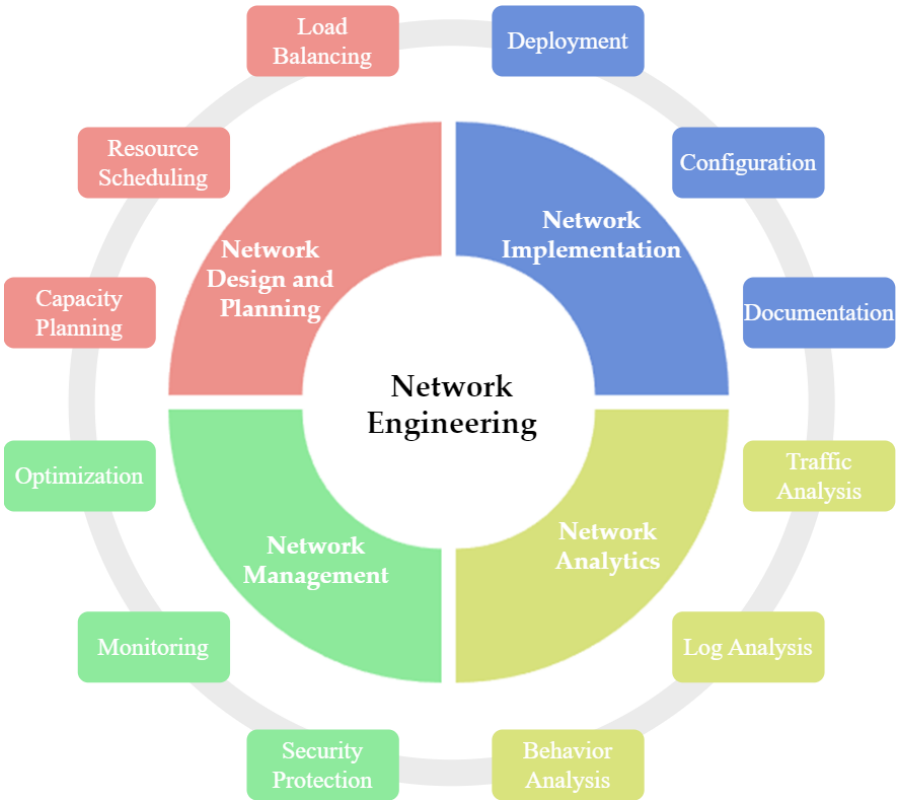


Figure 1. The comprehensive life cycle of network systems, highlighting the four primary stages in network engineering: Network Design and Planning, Network Implementation, Network Analytics, and Network Management. Each stage involves critical tasks such as resource scheduling, deployment, traffic analysis, and security protection. These four network engineering stages are interconnected, ensuring seamless management of the network from initial planning to ongoing optimization and protection. Network intelligence is pivotal in automating processes and improving the efficiency, accuracy, and reliability of network tasks across all stages.

2.2. Large Language Models

A large language model (LLM) is an advanced computational model renowned for its ability to perform a wide range of NLP tasks such as general-purpose language generation, text classification, and sentiment analysis. LLMs are built upon sophisticated language models and acquire their capabilities by learning intricate statistical relationships from extensive corpora of text. The training process for LLMs is both computationally intensive and data-rich, involving self-supervised and semi-supervised learning techniques. During this process, the models are exposed to vast datasets, enabling them to learn diverse linguistic patterns and contextual nuances. LLMs are particularly effective in text generation, a prominent application of GenAI. They operate by taking an initial input text and recursively predicting the next word or token, thereby constructing coherent and contextually appropriate sentences and paragraphs. This predictive mechanism enables LLMs to generate human-like text that is not only grammatically correct but also contextually relevant and nuanced. The versatility of LLMs extends to various NLP applications such as machine translation, which is converting text from one language to another, and text summarization, which is condensing long documents into concise summaries. The impact of LLMs has been profound, especially after being incorporated as a chat tool that can understand and respond to inquiries with high accuracy. This integration allows for real-time interactions, promptly addressing problems and significantly reducing the time and effort required by humans.

LLMs are advanced artificial neural networks that employ the transformer architecture, a revolutionary development in deep learning. Originally introduced in [78], the trans-

former model uses an encoder–decoder structure designed to handle sequence-to-sequence tasks efficiently. The transformer model architecture is shown in Figure 2. The encoder processes the input sequence, capturing relevant information, while the decoder generates the output sequence by attending to the output of the encoder and previously generated tokens. This dual structure enables the transformer to excel in NLP tasks, where understanding and generating sequences of text are critical. The mathematical foundation of this architecture involves self-attention mechanisms, which allow the model to dynamically weigh the importance of different words in a sentence. The encoder–decoder structure works by first passing the input sequence through the encoder layers, which consist of multi-head self-attention and feed-forward neural networks. The self-attention mechanism can be mathematically represented as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V,$$

where Q, K, and V are the query, key, and value matrices derived from the input embeddings, and d_k is the dimension of the keys. This mechanism allows the model to focus on different parts of the input sequence, capturing long-range dependencies and contextual relationships. In the decoder, a similar self-attention mechanism is used, but with an additional encoder–decoder attention layer that attends to the output of the encoder. This allows the decoder to generate the output sequence while considering the entire input sequence, enhancing the coherence and relevance of the generated text.

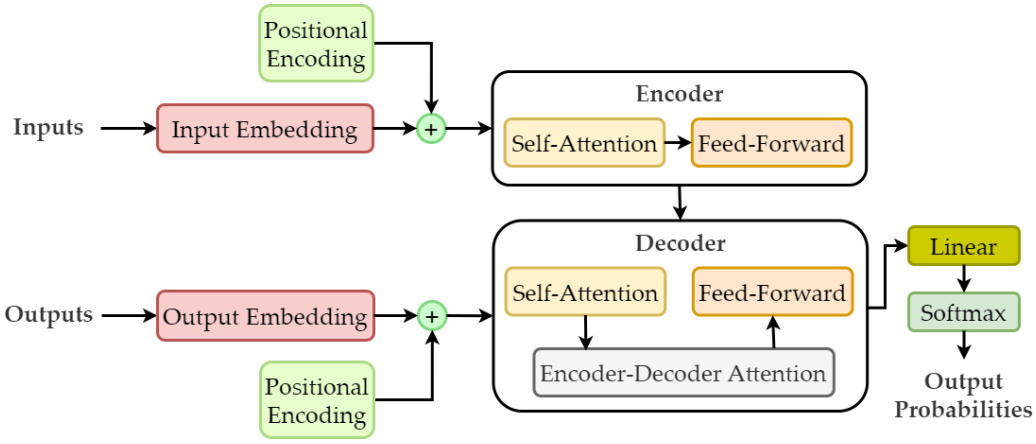


Figure 2. The traditional encoder–decoder transformer model architecture.

In recent years, the development of LLMs has accelerated, leading to the creation of numerous innovative models [28,79–126]. Notably, the largest and most capable LLMs today (as of June 2024) are built using a decoder-only transformer-based architecture. For instance, the GPT series from OpenAI and the LLaMA models from Meta AI are notable examples of decoder-only transformer models that have pushed the boundaries of what LLMs can achieve. The streamlined design of this architecture focuses solely on the generative aspect, making it particularly effective for tasks involving large-scale text generation. The decoder-only model omits the encoder, relying instead on self-attention to handle both the input context and the generation process. This architecture simplifies the model, allowing for more efficient processing and the handling of extensive datasets, which is crucial for training models with billions of parameters. In Figure 3, we show the timeline of the development of LLMs since the introduction of the GPT-1 model, highlighting the rapid evolution and advancements in LLMs.

The remarkable generalization power of LLMs allows them to perform a wide range of tasks. However, this strength also introduces a notable limitation: LLMs often produce responses that are too general when applied to domain-specific tasks. Their broad training on extensive datasets means they may lack the precision required for specialized fields.

Traditionally, fine-tuning has been the primary method for adapting LLMs to specific tasks. This process involves retraining the model on a smaller, domain-specific dataset to improve its performance in particular areas. More recently, the advent of larger models has shown that similar or even superior results can be achieved through prompt engineering. This technique involves crafting specific input prompts to guide the responses of the model, effectively steering the general-purpose capabilities of the LLM towards more targeted outputs. To address the hallucination problem, where models generate plausible but incorrect or nonsensical information, and the issue of inheriting inaccuracies and biases from training datasets, retrieval-augmented generation (RAG) has been developed. RAG incorporates external retrieval mechanisms into LLMs to access domain-specific knowledge bases, enhancing the ability of the model to generate accurate, up-to-date, and relevant responses in specialized fields [127,128]. Various applications have been developed using these three approaches to enhance LLMs for domain-specific purposes. Despite these advancements, ensuring that LLMs provide precise and reliable information for domain-specific tasks remains an ongoing area of research. This study explores the application of LLMs in solving problems within the network domain. We examine the potential of LLMs to enhance network engineering, highlighting both the capabilities and the challenges that need to be addressed to fully realize the potential of LLMs in this field.

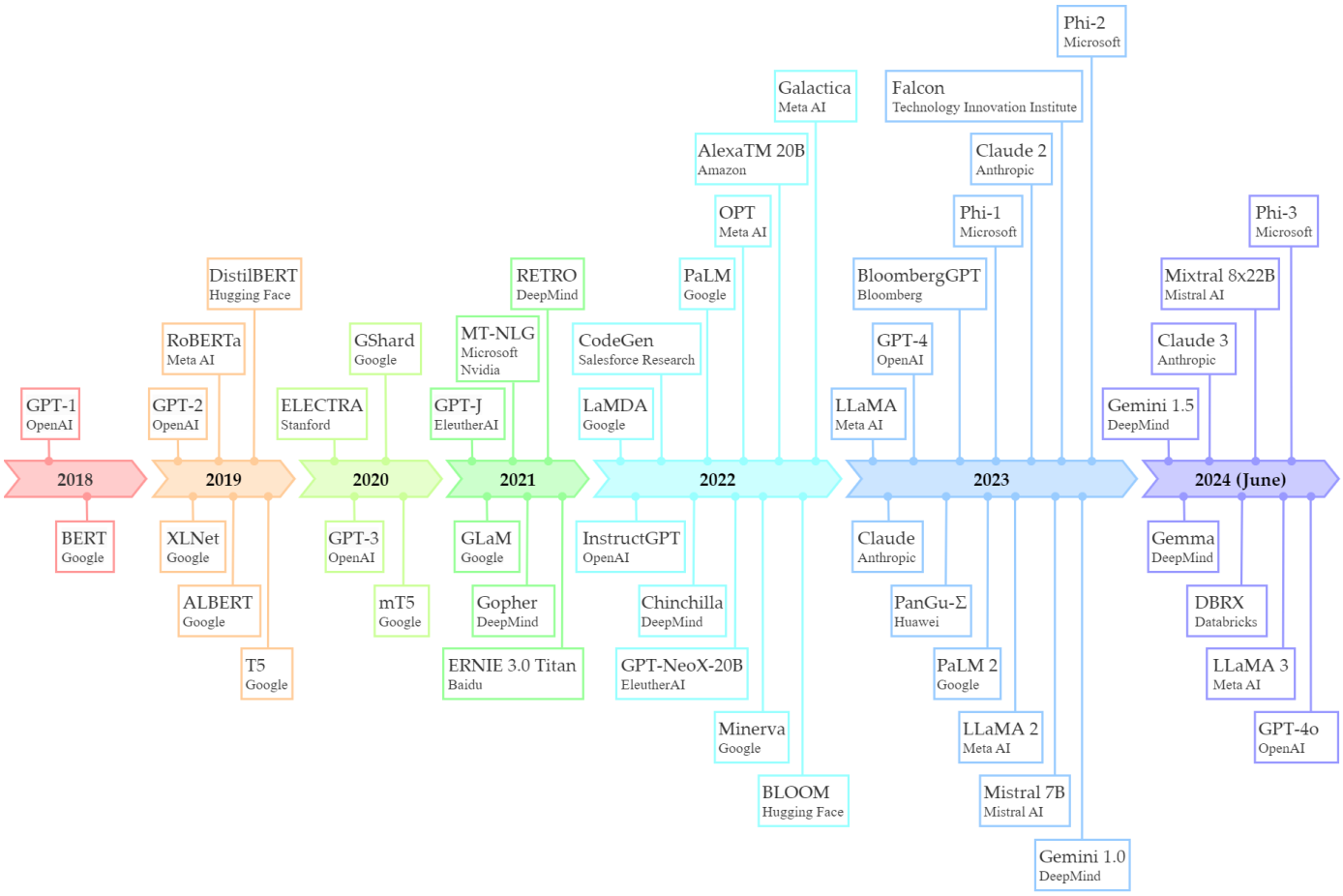


Figure 3. The timeline of the development of LLMs from 2018 to 2024 (June), showcasing key advancements and notable models.

2.3. From Natural Language to Network Configuration Language

Natural language and network configuration language differ significantly in structure and complexity, presenting unique challenges in bridging the two for effective network operations. Natural language is inherently flexible, context-rich, and user-friendly, designed to accommodate human communication in a fluid and dynamic manner. In contrast,

network configuration language is highly technical, consisting of non-standardized formats, symbols, and domain-specific terminologies. It encompasses a wide range of elements from high-level management policies to low-level technical specifics such as the command line interface and the network access control list. Traditional methods for translating natural language into network commands are limited to formalized approaches such as entity abstraction and template filling. These methods can only provide standardized translations, which are often insufficient for the nuanced and complex requirements of network operations. Network configuration language includes not only specialized nouns and protocols but also strict rules and mathematical constraints that govern its functionality. This complexity makes it challenging for non-experts to interact with network systems effectively using conventional translation methods.

Existing modern network paradigms such as intent-based networking (IBD), software-defined networking (SDN), and goal-oriented (GO) communication for networking are essential techniques that leverage natural language for network operations. IBD allows users to specify their network requirements in natural language terms, which are then automatically translated into technical configurations. SDN separates the control plane from the data plane, enabling more flexible and dynamic management of network resources through high-level commands. GO communication focuses on achieving specific network objectives defined in natural language, streamlining the process of network configuration and management. By leveraging advanced techniques and frameworks, we can effectively interchange natural language and network configuration language, creating more intuitive and efficient network management systems. This approach enables even non-experts to perform complex network tasks with ease and confidence. In Table 2, we illustrate how natural language commands can be translated into specific network operations using various features and examples. By translating high-level natural language inputs into detailed network configuration tasks, we can simplify complex network management, making it accessible to users without deep technical expertise.

Table 2. The examples of translating natural language commands into specific network operations using GPT-4.

Natural Language Prompt	Network Operation	Features	Example
Distribute incoming traffic evenly across three servers to ensure optimal performance.	Load Balancing Configuration	Distribute traffic using a 4-tuple (source Internet Protocol (IP), destination IP, source port, destination port) and utilize unique identifiers to avoid traffic polarization, supporting both IPv4 and IPv6 headers.	`ip cef load-sharing algorithm include-ports source destination gtp`
Configure the router to prioritize voice traffic over other types of traffic.	QoS Configuration	Prioritize specific types of traffic by applying pre-defined policies to network interfaces.	`interface GigabitEthernet0/1 service-policy output PRIORITIZE-VOICE`
Analyze the firewall logs to identify any failed login attempts.	Log Analysis	Filter and search through log entries based on criteria.	`grep ‘‘Failed login’’ /var/log/firewall.log`
Block all external traffic from accessing the internal server located at 198.168.100.2 to enhance security.	Firewall Rule Configuration	Establish access control by defining rules to block unauthorized external traffic to specific internal server IP addresses.	`deny ip any host 198.168.100.2`

The disparity between the conversational nature of natural language and the precise, rule-bound nature of network configuration language necessitates advanced solutions to seamlessly bridge these two forms of communication, translating user intents into accurate network commands. LLMs have the potential to significantly enhance network quality of experience (QoE) by offering customized responses to specific natural language inputs from users. These models can interpret the nuanced requirements conveyed through natural language and convert them into precise network operations. Techniques such as fine-tuning, prompt engineering, and RAG play crucial roles in this process. For example, LLMs can collaborate with IBN by fine-tuning to align with specific network policies and protocols, enabling the accurate translation of high-level user intents into precise network configurations. In SDN, prompt engineering can be used to create specific input prompts that help LLMs generate dynamic control commands, adjusting network resources in real time based on user requirements. For GO communication, RAG can integrate external knowledge bases, allowing LLMs to access the latest domain-specific information and optimize network performance to meet specific objectives, such as minimizing latency or maximizing throughput. These advanced techniques, which translate natural language into network configuration language, collectively enhance the precision, responsiveness, and adaptability of network management systems.

3. Large Language Models for Networking

In this section, we discuss research focused on the applications of LLMs across the four primary stages of network engineering. Table 3 summarizes the discussed works. Note that some works may pertain to more than one stage of network engineering. Therefore, we categorize each work based on the primary keywords and key objectives, although they may relate to multiple stages.

3.1. Network Design and Planning

The stage of network design and planning involves developing efficient and robust network architectures tailored to specific performance, security, and scalability needs. This process includes tasks such as topology design, resource allocation, and capacity planning. Incorporating LLMs into network design and planning can streamline these tasks by utilizing their advanced language comprehension and inference capabilities to generate innovative and optimized solutions. For example, NetLLM in [129] leverages pre-trained knowledge and powerful inference abilities to solve various networking tasks, achieving better performance and generalization than traditional methods. In [130], the authors propose using multi-agent generative LLMs in wireless networks to develop intelligent, self-governed networks that facilitate collaborative decision-making at the edge. Similarly, an LLM-based resource scheduling system using three LLM agents is proposed in [131] to translate user voice requests into resource allocation vectors for personalized optimization in wireless and digitalized energy networks, aiming to enhance user-centric approaches through natural language interfaces. llmQoS [132] utilizes LLMs to extract information from natural language descriptions of web users and services, which is then combined with historical QoS data to enhance web service recommendations and QoS predictions. A cloud load balancing framework in [133] integrates reinforcement learning, LLMs, and edge intelligence to enhance performance, security, scalability, and operational efficiency for dynamic cloud environments. NetGPT [134], a collaborative cloud–edge framework, enhances personalized generative services and intelligent network management by combining LLMs with location-based information at the edge.

Table 3. A summary of selected works on LLMs for networking.

Network Process	Work	Year	Baseline Model(s)	Network Application(s)
Network Design and Planning	Desai et al. [133]	2023	GPT-4	Load Balancing, Capacity Planning
	Zou et al. [130]	2023	GPT-4	Intent-Based Networking, Telecommunications, 6G, Network Energy Saving
	NetLLM [129]	2024	LLaMA 2	QoE, Bandwidth Management, Job Scheduling, Adaptive Bitrate Streaming
	NetGPT [134]	2024	GPT-2, LLaMA	QoS, Network Provisioning, Cloud-Edge Resource Allocation and Scheduling
	llmQoS [132]	2024	RoBERTa, Phi-3	QoS, Network Service Recommendation
	Mongaillard et al. [131]	2024	LLaMA 3, GPT-4, Gemini 1.5	QoE, Power Scheduling, Resource Allocation
Network Implementation	VPP [135]	2023	GPT-4	Router Configuration
	Ciri [136]	2023	GPT-4, GPT-3.5, Claude 3, Code Llama, DeepSeek	Configuration Validation
	NETBUDDY [137]	2023	GPT-4	Network Configuration
	Emergence [138]	2023	GPT-4, GPT-3.5	Intent-Based Networking, Virtual Network Function, Network Policy
	GPT-FL [139]	2023	GPT-3	Federated Learning
	LP-FL [140]	2023	BERT	Federated Learning
	ChatAFL [141]	2024	GPT-3.5	Cybersecurity, Network Protocol
	Mekrache et al. [142]	2024	Code Llama	Network Configuration, Intent-Based Networking, Next Generation Network, Network Service Descriptor
	GeNet [143]	2024	GPT-4	Network Configuration, Network Topology, Intent-Based Networking
	S-Witch [144]	2024	GPT-3.5	Network Configuration, Intent-Based Networking, Network Digital Twin
	Mekrache et al. [145]	2024	Code Llama	Network Configuration, Intent-Based Networking, Next Generation Network
	Fuad et al. [146]	2024	GPT-4, GPT-3.5, LLaMA 2, Mistral 7B	Network Configuration, Intent-Based Networking
Network Analytics	NetBERT [147]	2020	BERT	Networking Text Classification and Networking Information Retrieval
	GPT-2C [148]	2021	GPT-2	Log Analysis, Intrusion Detection
	NTT [149]	2022	Vanilla Transformer	Network Dynamics, Network Traffic
	LogGPT [150]	2023	GPT-3.5	Log Analysis, Anomaly Detection
	LAnoBERT [151]	2023	BERT	Log Analysis, Anomaly Detection
	NetLM [152]	2023	GPT-4	Telecommunication, Network Traffic, Intent-Based Networking

Table 3. *Cont.*

Network Process	Work	Year	Baseline Model(s)	Network Application(s)
Network Analytics	BERTOps [153]	2023	BERT	Log Analysis
	LogGPT [154]	2023	GPT-2	Log Analysis, Anomaly Detection
	Szabó et al. [155]	2023	GPT-3.5, GPT-4	Cybersecurity, Vulnerability Detection
	Piovesan et al. [156]	2024	Phi-2	Telecommunication
	LILAC [157]	2024	GPT-3.5	Log Parsing, Log Analysis
	Mobile-LLaMA [158]	2024	LLaMA 2	Network Data Analytic Function, Telecommunications, 5G
Network Management	Wong et al. [159]	2020	DistilBERT	Cybersecurity, Man-in-the-Middle Attack, Internet of Things
	CyBERT [160]	2021	BERT	Cybersecurity
	MalBERT [161]	2021	BERT	Cybersecurity, Malware Detection
	SecureBERT [162]	2022	BERT	Cybersecurity, Cyber Threat Intelligence
	Demırcı et al. [163]	2022	GPT-2	Cybersecurity, Malware Detection
	NorBERT [164]	2022	BERT	Network Monitoring, Fully Qualified Domain Name
	PAC-GPT [165]	2023	GPT-3	Cybersecurity, Network Traffic
	Hamadanian et al. [166]	2023	GPT-4	Network Incident Management
	Owl [167]	2023	Vanilla Transformer	Information Security, Log Parsing, Anomaly Detection
	Mani et al. [168]	2023	GPT-4, GPT-3, Text-davinci-003, Google Bard	Network Lifecycle Management, Network Traffic, Program Synthesis
	Bariah et al. [169]	2023	GPT-2, BERT, DistilBERT, RoBERTa	QoS, Telecommunication
	Tann et al. [170]	2023	GPT-3.5, PaLM 2, Prometheus	Cybersecurity
	Cyber Sentinel [171]	2023	GPT-4	Cybersecurity
	Moskal et al. [172]	2023	GPT-3.5	Cybersecurity
	Net-GPT [173]	2023	LLaMA 2, DistilGPT-2	Cybersecurity, Network Protocol, Man-in-the-Middle Attack
	Sarabi et al. [174]	2023	RoBERTa	Network Measurement, Internet of Things
	HuntGPT [175]	2023	GPT-3.5	Cybersecurity, Anomaly Detection, Intrusion Detection
	Zhang et al. [176]	2023	GPT-2	Cybersecurity
	ShieldGPT [177]	2024	GPT-4	Cybersecurity, Network Traffic, Distributed Denial of Service Attack
	SecurityBERT [178]	2024	BERT	Cybersecurity, Cyber Threat Detection, Internet of Things
	Habib et al. [179]	2024	ALBERT	Network Optimization, Intent-Based Networking
	DoLLM [180]	2024	LLaMA 2	Cybersecurity, Distributed Denial of Service Attack

3.2. Network Implementation

Network implementation focuses on the deployment and configuration of network infrastructure, ensuring that network policies and designs are accurately executed. LLMs can enhance this phase by automating configuration tasks, translating high-level policies into actionable commands, and providing robust validation mechanisms. This leads to more precise and efficient implementation processes. For configuration, several innovative frameworks have been introduced. Verified Prompt Programming (VPP) in [135] combines GPT-4 with verifiers to automatically correct errors in router configurations. S-Witch in [144] uses a LLM combined with a network digital twin to generate and verify command line interface commands for commercial switches based on natural language requests. NETBUDDY in [137] utilizes LLMs to translate high-level network policies into low-level network configurations. Ciri [136], a generic LLM-based configuration validation framework, employs effective prompt engineering and few-shot learning to address hallucination and nondeterminism in LLMs during configuration validation. Additionally, the work in [142] proposes an LLM-based intent translation system that converts user-defined business network requirements expressed in natural language into Network Service Descriptors. GeNet [143], a multimodal copilot framework, leverages LLMs to streamline enterprise network design workflows by interpreting and updating network topologies and device configurations based on user intents.

Other significant contributions include an LLM-centric intent lifecycle management architecture for configuring and managing network services using natural language [145]. The study in [146] utilizes LLMs to automate network configurations through IBN, ensuring data privacy and configuration integrity. Emergence [138,181], an intent-based management system, employs the few-shot capability of LLMs with a Policy-based Abstraction Pipeline to create a closed control loop for intent deployment and application management. ChatAFL [141], an LLM-guided protocol implementation fuzzing engine, generates machine-readable information about protocols for enhanced security testing and coverage. In the context of federated learning, GPT-FL in [139] is a generative pre-trained model-assisted federated learning framework that leverages synthetic data generated by LLMs to train downstream models on the server, which are then fine-tuned with private client data. LP-FL in [140] combines few-shot prompt learning from LLMs with efficient communication and federating techniques, using Low-Rank Adaptation to reduce computation and communication costs and enabling iterative soft-label assigning to expand labeled data sets during the federated learning process.

3.3. Network Analytics

Network analytics is crucial for monitoring, analyzing, and understanding network performance, behavior, and security. It ensures optimal operation and timely detection and resolution of issues. By integrating LLMs into network analytics, we can leverage their powerful language understanding and inference capabilities to offer advanced solutions for real-time data analysis and predictive maintenance. Various approaches have been proposed for using LLMs in log analysis. For instance, GPT-2C in [148] is a run-time system that uses a fine-tuned GPT-2 model to parse dynamic logs for intrusion detection systems. LILAC in [157] leverages LLMs with an adaptive parsing cache to improve accuracy and efficiency in parsing complex log data for network analysis tasks. LogGPT [150], on the other hand, utilizes language interpretation capabilities of ChatGPT to tackle high-dimensional and noisy log data for anomaly detection in network management. LAnoBERT in [151] uses the BERT model for unsupervised log anomaly detection, enhancing performance in network log data analysis. Another variant of LogGPT in [154] employs GPT for predicting the next log entry based on preceding sequences and uses a reinforcement learning strategy to improve anomaly detection. BERTOps in [153] is an LLM-based framework designed for automating network tasks like log format detection, classification, and parsing to enhance operational workflows.

Beyond log analysis, other notable contributions include a GPT-based framework for detecting vulnerabilities in source code by analyzing sensitive code segments [155]. The Network Traffic Transformer (NTT) in [149] is adapted to learn and generalize network traffic dynamics from packet traces, showing potential for prediction tasks in networking. NetLM in [152] utilizes a transformer-based LLM for understanding and managing network traffic dynamics through multi-modal representation learning and incremental control policy generation. NetBERT in [147] is a domain-specific LLM pre-trained on networking corpora, improving tasks like classification and information retrieval within the networking domain. The study in [156] enhances Phi-2 with RAG to improve operational efficiency in telecom-related queries while addressing resource constraints. Mobile-LLaMA in [158] is a variant of the LLaMA 2 model tailored for 5G network management, enhancing capabilities in packet analysis, IP routing analysis, and performance analysis.

3.4. Network Management

Network management involves the continuous monitoring, control, and optimization of network performance, security, and reliability. LLMs provide advanced capabilities for automating and enhancing these tasks. Their sophisticated language processing power allows for improved operational efficiency, proactive security measures, and dynamic adjustments to network conditions, ultimately leading to more resilient and adaptive network systems. For cybersecurity, Cyber Sentinel in [171] is a task-oriented cybersecurity dialogue system leveraging chained GPT-4 models and prompt engineering to explain potential threats and take proactive security actions, enhancing transparency and decision-making in network management. SecureBERT in [162] is a cybersecurity language model trained on a large corpus of cybersecurity text to automate critical tasks in Cyber Threat Intelligence by capturing text connotations and transforming natural language text into machine-readable formats. The work in [170] evaluates the effectiveness of LLMs in solving cybersecurity Capture-The-Flag challenges. CyBERT [160], a domain-specific BERT model fine-tuned with a large corpus of cybersecurity data, is designed to enhance the performance of various cybersecurity-specific downstream tasks for Security Operations Centers by processing dense, fine-grained textual threat, attack, and vulnerability information. The work in [172] explores the use of LLMs in automating cyber campaigns, presenting a framework for a plan–act–report loop and prompt chaining to direct sequential decision processes in threat campaigns, highlighting the potential and ethical implications of LLMs in enhancing threat actor capabilities in network management. The study in [176] introduces a GPT-2-based approach to enhance network threat detection, aiming to improve the efficiency and accuracy of identifying and issuing early warnings for network threats. For malware detection, the work in [161] proposes using a BERT-based transformer architecture for automatically detecting and classifying malicious software in Android applications through static analysis of preprocessed source code features. The study in [163] proposes using LLMs for detecting malicious code by analyzing assembly instructions from Portable Executable files, evaluating their effectiveness in network management. For intrusion detection, HuntGPT [175] leverages the GPT-3.5 model as a conversational agent to enhance the explainability and user-friendliness of threat detection in network management.

For specific types of cyberattacks, research includes the introduction of a novel Man-in-the-Middle (MITM) attack scheme on Internet of Things (IoT) devices using the MQTT protocol, featuring an MQTT Parser and a BERT-based adversarial model to generate malicious messages, effectively evading various anomaly detection mechanisms in network management [159]. Net-GPT in [173] is an LLM-empowered offensive chatbot designed to launch Unmanned Aerial Vehicle (UAV)-based MITM attacks by mimicking network packets between UAVs and Ground Control Stations, leveraging fine-tuned LLMs on an edge server to enhance predictive accuracy and adaptability in network management. For Distributed Denial of Service (DDoS) attack, DoLLM in [180] utilizes open-source LLMs to analyze non-contextual network flows by projecting them into a semantic space of LLMs, enhancing the detection of Carpet Bombing DDoS attacks. ShieldGPT in [177]

is a comprehensive DDoS mitigation framework that leverages LLMs to detect attacks, represent network traffic, inject domain knowledge, and provide actionable mitigation measures through tailored prompt engineering and representation schemes. SecurityBERT [178], a novel architecture utilizing the BERT model for cyber threat detection in IoT networks, leverages a privacy-preserving encoding technique to achieve high accuracy and efficiency in identifying various attack types, making it suitable for deployment on resource-constrained IoT devices.

For general network management, PAC-GPT in [165] utilizes GPT-3 to generate synthetic data for enhancing machine learning methods in network flow and packet generation. The work in [166] proposes a holistic framework utilizing LLMs to improve incident management in network operations, analyzing fundamental requirements and future research directions based on insights from operators of a large public cloud provider. Owl [167], a specialized LLM for information technology (IT) operations, employs a mixture-of-adapter strategy for efficient tuning, demonstrating superior performance on IT-related tasks and benchmarks. The work in [168] employs LLMs to generate task-specific code from natural language queries, tackling issues related to explainability, scalability, and privacy for network operators in network management. NorBERT in [164] is a framework that adapts the BERT model from NLP to learn semantically meaningful embeddings for Fully Qualified Domain Names in communication networks, aiming to create deep models that generalize effectively across various network tasks and environments for passive network monitoring. The study in [169] proposes a framework for adapting LLMs to the telecom domain, aiming to automate network tasks to enhance intent-driven and self-evolving wireless networks. In [174], a transformer-based LLM is proposed for characterizing, clustering, and fingerprinting raw text from network measurements, generating robust embeddings for downstream tasks and identifying new IoT devices and server products through automated analysis and labeling of Internet scan data. The work in [179] utilizes a lightweight LLM to process intents, validating them against future traffic profiles and deploying machine learning-based network optimization applications to enhance key performance indicators.

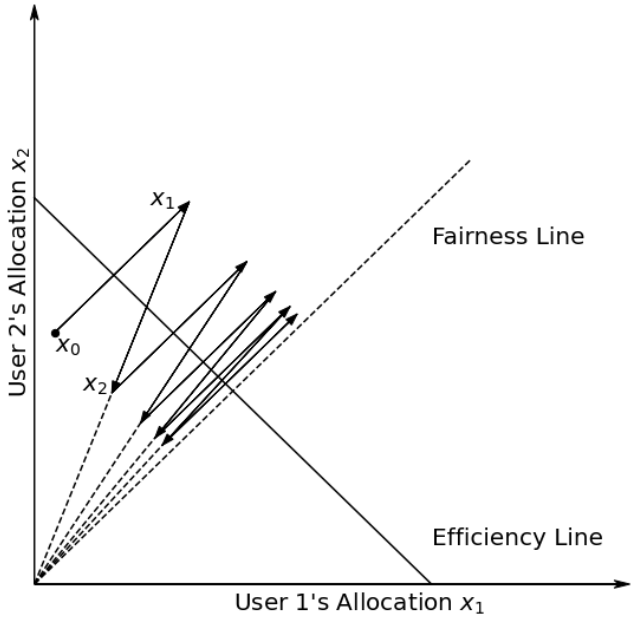
4. Open Challenges and Opportunities

Many challenges and opportunities remain in deploying LLMs for networking. In this section, we discuss some of these open challenges and opportunities.

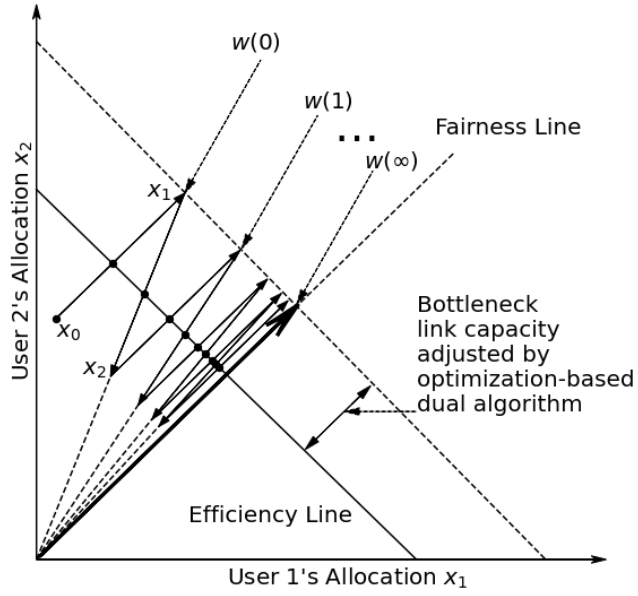
4.1. Congestion Control with Large Language Models

Historically, congestion control in network systems has relied heavily on mathematical models to manage data traffic and prevent network overloads [182–186]. These models provide an initial approximation of the actual network behavior, which can be exceedingly complex to analyze directly. Over time, several sophisticated and effective models, such as queuing theory and stochastic processes, have been developed. For network engineers, selecting the appropriate congestion control model often involves balancing mathematical simplicity with the fidelity of the model to real-world network dynamics. As network systems have become increasingly complex, identifying and applying the right model has become more critical. One of the most impactful approaches to congestion control is the Additive-Increase/Multiplicative-Decrease (AIMD) algorithm [187]. AIMD works by gradually increasing the data transmission rate until packet loss occurs, which signals network congestion. Upon detecting packet loss, the algorithm significantly reduces the transmission rate to alleviate congestion. This cycle of gradually increasing and then sharply decreasing the transmission rate helps maintain a balance between network utilization and congestion, making AIMD a cornerstone in traditional congestion control methods. The ingenuity behind the AIMD algorithm can be appreciated through a classical illustration of two flows, as shown in Figure 4a. This visual representation of the AIMD algorithm demonstrates the convergence of the rates of two users to the fairness line [187,188]. When the rate iterations are below the efficiency line, the additive-increase mechanism appears as a

45-degree increment plot. Above the efficiency line, the multiplicative-decrease mechanism appears as a slide to the midpoint between the current rate iteration and the origin.



(a)



(b)

Figure 4. Iterative balancing of resource allocations using AIMD. (**a**) A visual representation of the AIMD algorithm. (**b**) An illustration of the convergence of AIMD resource allocation based on Perron-Frobenius theory. The bold arrow represents the Perron-Frobenius right eigenvector of a non-negative matrix, illustrating its convergence towards the fairness line.

The AIMD algorithm, foundational to TCP congestion control [189,190], not only plays a critical role in managing traffic flow but also connects deeply to optimization theory through the Perron-Frobenius theorem and eigenvector computation. In distributed systems, where resources like bandwidth are shared, the AIMD algorithm ensures fairness by balancing the data transmission rate among nodes. The Perron-Frobenius theorem, which deals with the spectral properties of non-negative matrices, provides a mathematical

framework for understanding how distributed systems converge to stable states [191]. Specifically, the dominant eigenvector characterizes the long-term behavior of such systems, including network congestion control. This connection helps explain why iterative adjustments of AIMD lead to fairness in resource allocation, as the system naturally converges to an equilibrium where no single node dominates the resources of networks. By leveraging these insights, modern approaches to congestion control go beyond manually tuned algorithms, using computational methods to synthesize optimal control strategies that balance throughput and delay in complex network environments. As illustrated in Figure 4b, the iterative adjustments shown by the arrows lead to optimal resource allocation as the system moves towards the fairness point through a dual optimization process. This ensures the system converges to a stable allocation guided by the convex optimization and eigenvector analysis. The bold arrow points to the direction where the system reaches optimal performance (maximum efficiency without congestion), representing the convergence towards fairness under dynamic network conditions.

TCP ex Machina in [192] introduces Remy, an innovative approach to congestion control. Utilizing computational power, Remy automatically generates algorithms based on predefined network assumptions and objectives. Unlike traditional TCP protocols that manually define endpoint reactions to congestion signals, Remy synthesizes control algorithms by specifying network conditions and performance goals. This method allows for creating distributed algorithms that can adapt to various network environments, optimizing throughput and minimizing delay without requiring intrusive changes to network infrastructure. Remy-generated algorithms offer a significant improvement over manually designed protocols, providing a more adaptable and efficient approach to congestion control in diverse network conditions. The success of the automation approach proposed in TCP ex Machina represents a groundbreaking step towards automating the process of optimal congestion control, highlighting the potential for integrating advanced intelligence with TCP congestion control mechanisms. Central to TCP ex Machina is the use of network objectives or performance goals set by network operators to optimize network performance alongside network conditions. These crucial elements can be effectively managed by leveraging the capabilities of LLMs. For instance, network operators may find it challenging to articulate clear and precise network objectives or performance goals. LLMs can analyze detailed descriptions provided by network operators to accurately identify and distill the main objectives and goals that can be realistically achieved. This automated identification process ensures that network performance targets are well-defined and aligned with the intentions of the operators. Additionally, network operators may lack comprehensive visibility into current network conditions. LLMs can analyze incident reports and operator logs to identify patterns and infer network conditions. By synthesizing this information, LLMs can suggest adjustments to network objectives or performance goals, providing a more dynamic and responsive approach to congestion control. This approach can potentially facilitate informed decision-making and optimized network performance before actual implementation, demonstrating a significant advancement in network optimization.

In satellite networks like Starlink [193], the behavior of AIMD must be adapted to account for the higher latencies inherent in satellite communication. In terrestrial networks, AIMD increases the transmission rate incrementally until packet loss occurs, then reduces the rate significantly. However, in satellite environments, packet loss is often caused by long propagation delays or changes in satellite positioning, not necessarily congestion. This misinterpretation by AIMD could lead to unnecessary reductions in transmission rates, resulting in suboptimal throughput for users. To address this, modifications to AIMD are necessary for satellite networks. One such approach, inspired by the work of Sally Floyd on HighSpeed TCP [194], could involve adjusting the AIMD parameters (the additive increase rate and multiplicative decrease factor) to be more tolerant of delay and packet loss caused by satellite dynamics rather than congestion. In this way, the algorithm would better suit the fluctuating bandwidth and latency conditions found in satellite-to-ground communications of Starlink.

Automated machine learning (AutoML) and LLMs can further enhance the adaptability of AIMD in these challenging environments. AutoML can automatically fine-tune the parameters of AIMD by analyzing real-time network conditions, such as latency, throughput, and packet loss rates, and dynamically adjusting the behavior of the algorithm. This enables more efficient and resilient congestion control mechanisms that are responsive to the unique demands of satellite-based internet services. For example, AutoML models could analyze historical Starlink network performance data and predict the optimal AIMD settings for various traffic scenarios, ensuring better network performance across diverse geographic regions and environmental conditions. Additionally, the models could continuously learn from live network data to optimize AIMD parameters on the fly, significantly improving throughput and reducing latency even as Starlink satellites move in and out of range. Meanwhile, LLMs can complement this approach by providing intelligent recommendations and insights to network operators. By analyzing large datasets of Starlink network traffic, LLMs can help identify patterns in packet loss, latency, and bandwidth fluctuations that might not be immediately apparent. This can inform adjustments to AIMD settings or even suggest new variations of congestion control algorithms tailored to satellite networks. LLMs can also facilitate better human-AI collaboration by translating network insights into natural language, allowing operators to make informed decisions more easily. As such, while traditional AIMD congestion control mechanisms face limitations in satellite environments like Starlink, integrating AutoML and LLMs can significantly improve their overall performance and reliability.

In the past, around the 1980s, end users had no visibility into the underlying network conditions, making it difficult to optimize the AIMD algorithm effectively. Back then, network environments were simpler, and the performance of the AIMD algorithm depended largely on static assumptions. However, with the advent of more complex systems such as 5G, 6G, and satellite-based internet of SpaceX, optimizing network resource allocation even with AIMD has become increasingly challenging due to the dynamic and intricate nature of these networks. Emerging technologies like satellite-based non-terrestrial networks and Open Radio Access Network architecture in future wireless networks [195], further complicate the landscape, requiring more adaptive congestion control solutions. TCP Machina represents a significant advancement in the field of congestion control, utilizing supervised learning to automate the design of congestion control algorithms. By training on network simulations, TCP Machina is able to optimize congestion control strategies for various network scenarios, learning the most effective methods for resource allocation. However, while this approach offers improvements, it is limited by its reliance on predefined conditions and lacks the ability to dynamically adapt in real time to the constantly changing environments found in modern networks. The integration of LLMs and network log analysis introduces a new dimension of adaptability to this challenge. By analyzing network logs, LLMs can infer network conditions that are typically opaque to end users. Based on these insights, LLMs can dynamically adjust the parameters of the AIMD algorithm, specifically fine-tuning α (Additive Increase) to optimize resource allocation in response to evolving network conditions, and β (Multiplicative Decrease) to efficiently manage congestion. This approach allows for continuous, data-driven refinement of AIMD parameters, ensuring that network resource allocation is precisely tailored to the unique and fluctuating conditions of complex networks. In doing so, LLMs provide a level of responsiveness and efficiency in congestion control that surpasses static, scenario-based methods, making them particularly well-suited for the demands of modern networks. This approach utilizes unsupervised learning, offering greater flexibility and adaptability to unanticipated network conditions as compared to supervised learning methods.

4.2. Language Server Protocol with Large Language Models

The Language Server Protocol (LSP) is a critical standard in modern software development, designed to decouple the language-specific features of development environments from the editors or Integrated Development Environments (IDEs) that use them. This

protocol enables the creation of a single language server that can provide features like auto-completion, go-to-definition, and linting to multiple IDEs, simplifying the development process and improving the consistency of language support across different tools. LSP works by allowing the language server to communicate with the development tools using a common protocol, ensuring that features like syntax highlighting and code completion are uniformly applied, regardless of the editor being used. This decoupling facilitates easier maintenance and enhancement of language features and enables developers to use their preferred tools without sacrificing functionality.

In AI-assisted programming [196,197], integrating LLMs with the LSP marks a significant advancement. In particular, LLMs can enhance LSP by providing more sophisticated and context-aware programming assistance. For instance, LLMs can offer intelligent code suggestions, automated documentation generation, and advanced error detection and correction, thereby significantly boosting productivity and reducing time spent on routine coding tasks. Integrating LLMs with LSP can also improve the handling of networking protocols and configurations within development environments. Networking tasks often require precise and complex configurations, such as those involved with TCP/IP stacks. By embedding LSP within the networking stack, LLMs can assist in configuring and debugging network protocols, making it easier for developers to manage networking tasks. This integration has the potential to transform network programming by providing developers with intuitive tools that streamline building and maintaining network applications.

The potential for LLMs to enhance LSP goes beyond mere coding assistance. With their ability to understand and generate human-like text, LLMs can facilitate better communication and documentation within development teams. These models can automatically generate detailed explanations and summaries of code, making it easier for team members to understand complex implementations and collaborate more effectively.

4.3. Network Engineering Optimization for Large Language Models

Beyond leveraging LLMs to optimize network performance, another promising and important direction is exploring how network engineering strategies can optimize the functioning and deployment of LLMs. As LLMs are integrated into various applications, several challenges arise that require innovative network engineering solutions.

A significant challenge is managing high-demand LLM inference services that must handle a wide range of requests, from short chat conversations to lengthy document analysis, while ensuring fair processing for all clients. Traditional rate limiting can lead to under-utilization of resources and poor client experiences when spare capacity is available. The unpredictable request lengths and unique batching characteristics of LLMs on parallel accelerators further complicate this issue. To address this, the work in [198] introduces a new concept of LLM serving fairness based on a cost function that accounts for the number of input and output tokens processed and proposes the Virtual Token Counter scheduler to maintain fairness. Fairness is a critical aspect in network resource allocation, ensuring that all clients receive equitable access to computing resources [199–203]. A potential future direction is to integrate AIMD principles with scheduling algorithms to dynamically adjust resource allocation for LLMs based on real-time network conditions and usage patterns, potentially improving both fairness and resource utilization for LLMs.

Efficiently running LLM inference tasks on resource-limited mobile terminals is another critical challenge. Traditional deep reinforcement learning methods used for offloading these tasks to servers face issues such as data inefficiency and insensitivity to latency requirements. In [204], the authors propose an active inference algorithm with rewardless guidance for making offloading decisions and resource allocations in cloud–edge networks. This approach helps manage the distribution of LLM inference tasks between mobile devices and servers more effectively, ensuring that LLMs can be utilized efficiently even in resource-constrained environments. Incorporating network engineering techniques, such as optimizing data transmission paths and dynamically adjusting resource allocation, can

potentially enhance the efficiency and responsiveness of LLM-based applications on mobile platforms by improving data utilization efficiency and adaptability to changing task loads.

Creating new LLMs or fine-tuning existing ones for domain-specific tasks requires extensive data. The term "large" in LLM refers to models with billions of parameters. Deploying such parameter-heavy LLMs is resource-intensive, necessitating carefully designed hardware platforms. The work in [205] introduces the GenZ analytical tool to explore the relationship between LLM inference performance and various hardware design parameters. This study provides insights into configuring platforms for different LLM workloads and projecting future hardware capabilities needed for increasingly complex models. Network engineering can play a crucial role by optimizing data flow and reducing latency between distributed hardware components, ensuring that LLMs operate at peak efficiency across diverse environments.

Acquiring large, high-quality instruction data for training generative LLMs is often costly and difficult to access. The work in [206] proposes Federated Instruction Tuning (FedIT), which leverages federated learning to harness diverse instruction data generated by end users while addressing privacy concerns. This approach allows for the training of LLMs using data stored on local devices, thus enhancing their ability to generate authentic and natural responses. Future research could explore the integration of advanced network engineering techniques to optimize data synchronization and transmission in federated learning environments, ensuring efficient and secure data handling.

Interoperability among AI systems, particularly those involving LLMs, is a critical aspect of advancing AI applications [207]. The challenge lies in enabling seamless interaction and integration of diverse AI models and systems. One approach involves leveraging text-oriented exchange protocols to facilitate communication between different AI models, thereby enhancing the functionality and coherence of LLMs in various applications. Establishing semantic and syntactic standards for AI systems is essential. Such standards ensure that AI systems, including LLMs, can reliably exchange information, similar to how Internet protocols facilitate reliable data exchange across diverse systems. By adopting these standards, LLMs can operate more cohesively with other AI models and control systems, ensuring interoperability across different platforms and applications. Additionally, AI interoperability can promote cooperative interactions among various AI systems. This can be achieved through the development of standardized protocols that enable AI systems to share knowledge and insights, thereby improving their collective intelligence and decision-making capabilities. For LLMs, this means the ability to interact seamlessly with other AI models, enhancing their capacity to process and generate contextually accurate and relevant responses.

4.4. Challenges and Constraints of Implementing Large Language Models in Networks

LLMs rely on autoregressive mechanisms, generating text by predicting the next token (word or character) based on previous ones. This process allows them to generate coherent sequences of text that can be applied to various tasks, such as natural language understanding, translation, and even conversational AI. The autoregressive mechanism operates iteratively, resulting in highly fluent language models that can mimic human communication. However, deploying LLMs in network engineering presents challenges. While effective for NLP tasks, LLMs often lack the domain-specific knowledge required for complex networking tasks. Although LLMs can interpret inputs and generate responses, they rely on probabilistic associations learned from vast, general-purpose datasets, which can lead to linguistically accurate but contextually inappropriate responses for specific networking situations. In essence, LLMs do not possess the in-depth understanding of network infrastructure, protocols, and dynamics that come naturally to experienced human operators [208–210].

While LLMs are highly capable in general NLP tasks, their "intelligence" lacks human intuition and the ability to fully comprehend or anticipate edge cases that often arise in complex, real-world network environments. As LLMs are trained on data provided by

developers, they are confined to the patterns and information within those datasets. This means that when faced with scenarios or issues not present in their training data, LLMs might struggle to provide accurate or relevant solutions. For example, unforeseen network configurations, atypical network behaviors, or novel security threats might fall outside the learned scope of the model, leaving it ill-equipped to respond effectively. This limitation highlights a significant challenge in relying solely on LLMs for network operations. Since the models rely heavily on the completeness and diversity of the training data, gaps or biases in those data can lead to inaccurate or incomplete responses in real-time network operations. Additionally, in networking, where rapid and dynamic adjustments are often needed, LLMs may lack the adaptability to respond to entirely new conditions. This issue is particularly concerning in situations where real-time problem-solving or decision-making is critical, such as during unexpected outages or security breaches.

Given these constraints, LLMs are best suited to complement human operators rather than replace them. This points to the role of LLMs as part of a human-in-the-loop system for networking. Human expertise is essential for interpreting nuanced, context-specific situations, especially when conditions diverge from typical patterns seen in the training data. LLMs can assist by automating routine tasks, suggesting possible solutions, and providing rapid data-driven insights, but the ultimate decision-making and troubleshooting responsibilities still require human intervention to ensure network stability and reliability. Such an approach combines the efficiency and speed of LLM-driven automation with human expertise to ensure robust, adaptive, and contextually appropriate network management. LLMs can be highly beneficial in assisting non-experts or laypersons in understanding network configurations and troubleshooting steps, making network operations more accessible while still requiring human oversight for critical decision-making and problem resolution. This approach is crucial for leveraging the strengths of LLMs while compensating for their current limitations.

5. Conclusions

In this study, we provide an extensive review of the intersection between LLMs and next-generation networking technologies. By exploring the transformative role of LLMs in network design, implementation, analytics, and management, we highlight their potential to revolutionize traditional networking practices. The integration of LLMs into network systems presents both opportunities and challenges. Future research should focus on optimizing the deployment of LLMs in network environments, addressing issues related to computational efficiency, scalability, and interoperability. The development of domain-specific LLMs and the incorporation of advanced AI techniques like federated learning and reinforcement learning can further enhance the capabilities of network systems. Overall, this review underscores the importance of continuing to innovate and integrate AI technologies, particularly LLMs, into network engineering to build more resilient, efficient, and intelligent networks capable of meeting the demands of modern digital infrastructure.

Author Contributions: Conceptualization, C.-N.H. and C.-W.T.; methodology, C.-N.H., P.-D.Y., R.M. and C.-W.T.; validation, C.-N.H. and C.-W.T.; supervision, C.-N.H. and C.-W.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Singapore Ministry of Education Academic Research Fund (RG91/22 and NTU startup).

Data Availability Statement: No new data were created or analyzed in this study.

Acknowledgments: The authors are grateful for helpful discussions on the subject with D.-M. Chiu.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Shahraki, A.; Abbasi, M.; Piran, M.J.; Taherkordi, A. A comprehensive survey on 6G networks: Applications, core services, enabling technologies, and future challenges. *arXiv* **2021**, arXiv:2101.12475.
2. Salameh, A.I.; El Tarhuni, M. From 5G to 6G—Challenges, technologies, and applications. *Future Internet* **2022**, *14*, 117. [CrossRef]
3. Hossain, E.; Hasan, M. 5G cellular: Key enabling technologies and research challenges. *IEEE Instrum. Meas. Mag.* **2015**, *18*, 11–21. [CrossRef]
4. Haji, S.H.; Zeebaree, S.R.; Saeed, R.H.; Ameen, S.Y.; Shukur, H.M.; Omar, N.; Sadeeq, M.A.; Ageed, Z.S.; Ibrahim, I.M.; Yasin, H.M. Comparison of software defined networking with traditional networking. *Asian J. Res. Comput. Sci.* **2021**, *9*, 1–18. [CrossRef]
5. Hang, C.N.; Yu, P.D.; Chen, S.; Tan, C.W.; Chen, G. MEGA: Machine learning-enhanced graph analytics for infodemic risk management. *IEEE J. Biomed. Health Inform.* **2023**, *27*, 6100–6111. [CrossRef]
6. Hang, C.N.; Tsai, Y.Z.; Yu, P.D.; Chen, J.; Tan, C.W. Privacy-enhancing digital contact tracing with machine learning for pandemic response: A comprehensive review. *Big Data Cogn. Comput.* **2023**, *7*, 108. [CrossRef]
7. Suomalainen, J.; Juhola, A.; Shahabuddin, S.; Mämmelä, A.; Ahmad, I. Machine learning threatens 5G security. *IEEE Access* **2020**, *8*, 190822–190842. [CrossRef]
8. Chan, P.K.; Lippmann, R.P. Machine learning for computer security. *J. Mach. Learn. Res.* **2006**, *7*, 2669–2672.
9. Ahmed, M.; Mahmood, A.N.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. [CrossRef]
10. Zheng, L.; Cai, D.W.H.; Tan, C.W. Max-min fairness rate control in wireless networks: Optimality and algorithms by Perron-Frobenius theory. *IEEE Trans. Mob. Comput.* **2017**, *17*, 127–140. [CrossRef]
11. Zheng, L.; Hong, Y.W.P.; Tan, C.W.; Hsieh, C.L.; Lee, C.H. Wireless max-min utility fairness with general monotonic constraints by Perron-Frobenius theory. *IEEE Trans. Inf. Theory* **2016**, *62*, 7283–7298. [CrossRef]
12. Tan, C.W. Wireless network optimization by Perron-Frobenius theory. *Found. Trends Netw.* **2015**, *9*, 107–218. [CrossRef]
13. Tan, C.W. Optimal power control in Rayleigh-fading heterogeneous wireless networks. *IEEE/ACM Trans. Netw.* **2015**, *24*, 940–953. [CrossRef]
14. Zhang, T.; Qiu, H.; Mellia, M.; Li, Y.; Li, H.; Xu, K. Interpreting AI for networking: Where we are and where we are going. *IEEE Commun. Mag.* **2022**, *60*, 25–31. [CrossRef]
15. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 24824–24837.
16. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv* **2022**, arXiv:2203.11171.
17. Lyu, Q.; Havaldar, S.; Stein, A.; Zhang, L.; Rao, D.; Wong, E.; Apidianaki, M.; Callison-Burch, C. Faithful chain-of-thought reasoning. *arXiv* **2023**, arXiv:2301.13379.
18. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. Self-refine: Iterative refinement with self-feedback. *arXiv* **2023**, arXiv:2303.17651.
19. Yu, D.; Naik, S.; Backurs, A.; Gopi, S.; Inan, H.A.; Kamath, G.; Kulkarni, J.; Lee, Y.T.; Manoel, A.; Wutschitz, L.; et al. Differentially private fine-tuning of language models. *arXiv* **2021**, arXiv:2110.06500. [CrossRef]
20. Ziegler, D.M.; Stiennon, N.; Wu, J.; Brown, T.B.; Radford, A.; Amodei, D.; Christiano, P.; Irving, G. Fine-tuning language models from human preferences. *arXiv* **2019**, arXiv:1909.08593.
21. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.T.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Proceedings of the Advances in Neural Information Processing Systems, Online, 6–12 December 2020; Volume 33, pp. 9459–9474.
22. Jiang, Z.; Xu, F.F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; Neubig, G. Active retrieval augmented generation. *arXiv* **2023**, arXiv:2305.06983.
23. Hang, C.N.; Yu, P.D.; Tan, C.W. TrumorGPT: Query optimization and semantic reasoning over networks for automated fact-checking. In Proceedings of the 2024 58th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 13–15 March 2024; pp. 1–6.
24. Tan, C.W.; Chiu, D.M.; Lui, J.C.S.; Yau, D.K.Y. A distributed throttling approach for handling high bandwidth aggregates. *IEEE Trans. Parallel Distrib. Syst.* **2007**, *18*, 983–995. [CrossRef]
25. Singhal, K.; Azizi, S.; Tu, T.; Mahdavi, S.S.; Wei, J.; Chung, H.W.; Scales, N.; Tanwani, A.; Cole-Lewis, H.; Pfohl, S.; et al. Large language models encode clinical knowledge. *Nature* **2023**, *620*, 172–180. [CrossRef] [PubMed]
26. Thirunavukarasu, A.J.; Ting, D.S.J.; Elangovan, K.; Gutierrez, L.; Tan, T.F.; Ting, D.S.W. Large language models in medicine. *Nat. Med.* **2023**, *29*, 1930–1940. [CrossRef] [PubMed]
27. Laskar, M.T.R.; Alqahtani, S.; Bari, M.S.; Rahman, M.; Khan, M.A.M.; Khan, H.; Jahan, I.; Bhuiyan, A.; Tan, C.W.; Parvez, M.R.; et al. A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations. *arXiv* **2023**, arXiv:2407.04069.
28. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving language understanding by generative pre-training. *Preprint* 2018, *in press*.

29. Mata, J.; De Miguel, I.; Durán, R.J.; Merayo, N.; Singh, S.K.; Jukan, A.; Chamania, M. Artificial intelligence (AI) methods in optical networks: A comprehensive survey. *Opt. Switch. Netw.* **2018**, *28*, 43–57. [CrossRef]
30. Kibria, M.G.; Nguyen, K.; Villardi, G.P.; Zhao, O.; Ishizu, K.; Kojima, F. Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks. *IEEE Access* **2018**, *6*, 32328–32338. [CrossRef]
31. Xie, J.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Wang, C.; Liu, Y. A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. *IEEE Commun. Surv. Tutorials* **2018**, *21*, 393–430. [CrossRef]
32. Cayamcela, M.E.M.; Lim, W. Artificial intelligence in 5G technology: A survey. In Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 17–19 October 2018; pp. 860–865.
33. Zhao, Y.; Li, Y.; Zhang, X.; Geng, G.; Zhang, W.; Sun, Y. A survey of networking applications applying the software defined networking concept based on machine learning. *IEEE Access* **2019**, *7*, 95397–95417. [CrossRef]
34. Elsayed, M.; Erol-Kantarci, M. AI-enabled future wireless networks: Challenges, opportunities, and open issues. *IEEE Veh. Technol. Mag.* **2019**, *14*, 70–77. [CrossRef]
35. Chen, J.; Ran, X. Deep learning with edge computing: A review. *Proc. IEEE* **2019**, *107*, 1655–1674. [CrossRef]
36. Zhang, C.; Patras, P.; Haddadi, H. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 2224–2287. [CrossRef]
37. Sun, Y.; Peng, M.; Zhou, Y.; Huang, Y.; Mao, S. Application of machine learning in wireless networks: Key techniques and open issues. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 3072–3108. [CrossRef]
38. Wang, C.X.; Di Renzo, M.; Stanczak, S.; Wang, S.; Larsson, E.G. Artificial intelligence enabled wireless networking for 5G and beyond: Recent advances and future challenges. *IEEE Wirel. Commun.* **2020**, *27*, 16–23. [CrossRef]
39. Nguyen, D.C.; Cheng, P.; Ding, M.; Lopez-Perez, D.; Pathirana, P.N.; Li, J.; Seneviratne, A.; Li, Y.; Poor, H.V. Enabling AI in future wireless networks: A data life cycle perspective. *IEEE Commun. Surv. Tutorials* **2020**, *23*, 553–595. [CrossRef]
40. Semong, T.; Maupong, T.; Anokye, S.; Kehulakae, K.; Dimakatso, S.; Boipelo, G.; Sarefo, S. Intelligent load balancing techniques in software defined networks: A survey. *Electronics* **2020**, *9*, 1091. [CrossRef]
41. Zeydan, E.; Turk, Y. Recent advances in intent-based networking: A survey. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; pp. 1–5.
42. Mukhopadhyay, S.C.; Tyagi, S.K.S.; Suryadevara, N.K.; Piuri, V.; Scotti, F.; Zeadally, S. Artificial intelligence-based sensors for next generation IoT applications: A review. *IEEE Sensors J.* **2021**, *21*, 24920–24932. [CrossRef]
43. Chang, Z.; Liu, S.; Xiong, X.; Cai, Z.; Tu, G. A survey of recent advances in edge-computing-powered artificial intelligence of things. *IEEE Internet Things J.* **2021**, *8*, 13849–13875. [CrossRef]
44. Letaief, K.B.; Shi, Y.; Lu, J.; Lu, J. Edge artificial intelligence for 6G: Vision, enabling technologies, and applications. *IEEE J. Sel. Areas Commun.* **2021**, *40*, 5–36. [CrossRef]
45. Murshed, M.S.; Murphy, C.; Hou, D.; Khan, N.; Ananthanarayanan, G.; Hussain, F. Machine learning at the network edge: A survey. *ACM Comput. Surv.* **2021**, *54*, 1–37. [CrossRef]
46. Song, L.; Hu, X.; Zhang, G.; Spachos, P.; Plataniotis, K.N.; Wu, H. Networking systems of AI: On the convergence of computing and communications. *IEEE Internet Things J.* **2022**, *9*, 20352–20381. [CrossRef]
47. Gupta, C.; Johri, I.; Srinivasan, K.; Hu, Y.C.; Qaisar, S.M.; Huang, K.Y. A systematic review on machine learning and deep learning models for electronic information security in mobile networks. *Sensors* **2022**, *22*, 2017. [CrossRef] [PubMed]
48. Macas, M.; Wu, C.; Fuertes, W. A survey on deep learning for cybersecurity: Progress, challenges, and opportunities. *Comput. Netw.* **2022**, *212*, 109032. [CrossRef]
49. Salau, B.A.; Rawal, A.; Rawat, D.B. Recent advances in artificial intelligence for wireless internet of things and cyber–physical systems: A comprehensive survey. *IEEE Internet Things J.* **2022**, *9*, 12916–12930. [CrossRef]
50. Singh, R.; Gill, S.S. Edge AI: A survey. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 71–92. [CrossRef]
51. Zuo, Y.; Guo, J.; Gao, N.; Zhu, Y.; Jin, S.; Li, X. A survey of blockchain and artificial intelligence for 6G wireless communications. *IEEE Commun. Surv. Tutorials* **2023**, *25*, 2494–2528. [CrossRef]
52. Bourechak, A.; Zedadra, O.; Kouahla, M.N.; Guerrieri, A.; Seridi, H.; Fortino, G. At the confluence of artificial intelligence and edge computing in IoT-based applications: A review and new perspectives. *Sensors* **2023**, *23*, 1639. [CrossRef]
53. Gao, M. The advance of GPTs and language model in cyber security. *Highlights Sci. Eng. Technol.* **2023**, *57*, 195–202. [CrossRef]
54. Tarkoma, S.; Morabito, R.; Sauvola, J. AI-native interconnect framework for integration of large language model technologies in 6G systems. *arXiv* **2023**, arXiv:2311.05842.
55. Gill, S.S.; Golec, M.; Hu, J.; Xu, M.; Du, J.; Wu, H.; Walia, G.K.; Murugesan, S.S.; Ali, B.; Kumar, M.; et al. Edge AI: A taxonomy, systematic review and future directions. *arXiv* **2024**, arXiv:2407.04053.
56. Alhammadi, A.; Shayea, I.; El-Saleh, A.A.; Azmi, M.H.; Ismail, Z.H.; Kouhalvandi, L.; Saad, S.A. Artificial intelligence in 6G wireless networks: Opportunities, applications, and challenges. *Int. J. Intell. Syst.* **2024**, *2024*, 8845070. [CrossRef]
57. Ospina Cifuentes, B.J.; Suárez, Á.; García Pineda, V.; Alvarado Jaimes, R.; Montoya Benitez, A.O.; Grajales Bustamante, J.D. Analysis of the use of artificial intelligence in software-defined intelligent networks: A survey. *Technologies* **2024**, *12*, 99. [CrossRef]
58. Chen, Z.; Zhang, Z.; Yang, Z. Big AI models for 6G wireless networks: Opportunities, challenges, and research directions. *IEEE Wirel. Commun.* **2024**, *31*, 164–172. [CrossRef]

59. Ozkan-Ozay, M.; Akin, E.; Aslan, Ö.; Kosunalp, S.; Iliev, T.; Stoyanov, I.; Beloev, I. A comprehensive survey: Evaluating the efficiency of artificial intelligence and machine learning techniques on cyber security solutions. *IEEE Access* **2024**, *12*, 12229–12256. [CrossRef]
60. Celik, A.; Eltawil, A.M. At the dawn of generative AI era: A tutorial-cum-survey on new frontiers in 6G wireless intelligence. *IEEE Open J. Commun. Soc.* **2024**, *5*, 2433–2489. [CrossRef]
61. Khoramnejad, F.; Hossain, E. Generative AI for the optimization of next-generation wireless networks: Basics, state-of-the-art, and open challenges. *arXiv* **2024**, arXiv:2405.17454.
62. Bhardwaj, S.; Singh, P.; Pandit, M.K. A survey on the integration and optimization of large language models in edge computing environments. In Proceedings of the 2024 16th International Conference on Computer and Automation Engineering (ICCAE), Melbourne, Australia, 14–16 March 2024; pp. 168–172.
63. Karapantelakis, A.; Alizadeh, P.; Alabassi, A.; Dey, K.; Nikou, A. Generative AI in mobile networks: A survey. *Ann. Telecommun.* **2024**, *79*, 15–33. [CrossRef]
64. Zhou, H.; Hu, C.; Yuan, Y.; Cui, Y.; Jin, Y.; Chen, C.; Wu, H.; Yuan, D.; Jiang, L.; Wu, D.; et al. Large language model (LLM) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities. *arXiv* **2024**, arXiv:2405.10825. [CrossRef]
65. Mistry, H.K.; Mavani, C.; Goswami, A.; Patel, R. Artificial intelligence For networking. *Educ. Adm. Theory Pract.* **2024**, *30*, 813–821.
66. Martini, B.; Bellisario, D.; Coletti, P. Human-centered and sustainable artificial intelligence in industry 5.0: Challenges and perspectives. *Sustainability* **2024**, *16*, 5448. [CrossRef]
67. Barbosa, G.; Theeranantachai, S.; Zhang, B.; Zhang, L. A comparative evaluation of TCP congestion control schemes over low-Earth-orbit (LEO) satellite networks. In Proceedings of the 18th Asian Internet Engineering Conference, Bangkok, Thailand, 7–9 November 2023; pp. 105–112.
68. Roshan, K.; Zafar, A.; Haque, S.B.U. Untargeted white-box adversarial attack with heuristic defence methods in real-time deep learning based network intrusion detection system. *Comput. Commun.* **2024**, *218*, 97–113. [CrossRef]
69. Qiu, Y.; Ma, L.; Priyadarshi, R. Deep learning challenges and prospects in wireless sensor network deployment. *Arch. Comput. Methods Eng.* **2024**, *31*, 3231–3254. [CrossRef]
70. Khan, M.; Ghafoor, L. Adversarial machine learning in the context of network security: Challenges and solutions. *J. Comput. Intell. Robot.* **2024**, *4*, 51–63.
71. Priyadarshi, R. Exploring machine learning solutions for overcoming challenges in IoT-based wireless sensor network routing: A comprehensive review. *Wirel. Netw.* **2024**, *30*, 2647–2673. [CrossRef]
72. Ullah, F.; Ullah, S.; Srivastava, G.; Lin, J.C.W. IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. *Digit. Commun. Netw.* **2024**, *10*, 190–204. [CrossRef]
73. Latif, S.; Boulila, W.; Koubaa, A.; Zou, Z.; Ahmad, J. DTL-IDS: An optimized intrusion detection framework using deep transfer learning and genetic algorithm. *J. Netw. Comput. Appl.* **2024**, *221*, 103784. [CrossRef]
74. He, M.; Wang, X.; Wei, P.; Yang, L.; Teng, Y.; Lyu, R. Reinforcement learning meets network intrusion detection: A transferable and adaptable framework for anomaly behavior identification. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 2477–2492. [CrossRef]
75. Wu, G. Deep reinforcement learning based multi-layered traffic scheduling scheme in data center networks. *Wirel. Netw.* **2024**, *30*, 4133–4144. [CrossRef]
76. Kuo, C.Y.; Hang, C.N.; Yu, P.D.; Tan, C.W. Parallel counting of triangles in large graphs: Pruning and hierarchical clustering algorithms. In Proceedings of the 2018 IEEE High Performance extreme Computing Conference (HPEC), Waltham, MA, USA, 25–27 September 2018; pp. 1–6.
77. Hang, C.N.; Yu, P.D.; Tan, C.W. Parallel counting of subgraphs in large graphs: Pruning and hierarchical clustering algorithms. In *Online Social Networks: Perspectives, Applications and Developments*; Nova Science Publishers, Inc.: Hauppauge, NY, USA, 2020; pp. 141–164.
78. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
79. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
80. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
81. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. XLNet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5753–5763.
82. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
83. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
84. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
85. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.

86. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. ELECTRA: Pre-training text encoders as discriminators rather than generators. *arXiv* **2020**, arXiv:2003.10555.
87. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
88. Lepikhin, D.; Lee, H.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; Chen, Z. GShard: Scaling giant models with conditional computation and automatic sharding. *arXiv* **2020**, arXiv:2006.16668.
89. Xue, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; Raffel, C. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv* **2020**, arXiv:2010.11934.
90. Wang, B.; Komatsuzaki, A. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. 2021. Available online: https://huggingface.co/EleutherAI/gpt-j-6b (accessed on 23 September 2024).
91. Du, N.; Huang, Y.; Dai, A.M.; Tong, S.; Lepikhin, D.; Xu, Y.; Krikun, M.; Zhou, Y.; Yu, A.W.; Firat, O.; et al. GLaM: Efficient scaling of language models with mixture-of-experts. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 5547–5569.
92. Smith, S.; Patwary, M.; Norick, B.; LeGresley, P.; Rajbhandari, S.; Casper, J.; Liu, Z.; Prabhumoye, S.; Zerveas, G.; Korthikanti, V.; et al. Using DeepSpeed and Megatron to train Megatron-Turing NLG 530B, a large-scale generative language model. *arXiv* **2022**, arXiv:2201.11990.
93. Rae, J.W.; Borgeaud, S.; Cai, T.; Millican, K.; Hoffmann, J.; Song, F.; Aslanides, J.; Henderson, S.; Ring, R.; Young, S.; et al. Scaling language models: Methods, analysis & insights from training Gopher. *arXiv* **2021**, arXiv:2112.11446.
94. Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; Van Den Driessche, G.B.; Lespiau, J.B.; Damoc, B.; Clark, A.; et al. Improving language models by retrieving from trillions of tokens. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 2206–2240.
95. Wang, S.; Sun, Y.; Xiang, Y.; Wu, Z.; Ding, S.; Gong, W.; Feng, S.; Shang, J.; Zhao, Y.; Pang, C.; et al. ERNIE 3.0 Titan: Exploring larger-scale knowledge enhanced pre-training for language understanding and generation. *arXiv* **2021**, arXiv:2112.12731.
96. Thoppilan, R.; De Freitas, D.; Hall, J.; Shazeer, N.; Kulshreshtha, A.; Cheng, H.T.; Jin, A.; Bos, T.; Baker, L.; Du, Y.; et al. LaMDA: Language models for dialog applications. *arXiv* **2022**, arXiv:2201.08239.
97. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 27730–27744.
98. Nijkamp, E.; Pang, B.; Hayashi, H.; Tu, L.; Wang, H.; Zhou, Y.; Savarese, S.; Xiong, C. CodeGen: An open large language model for code with multi-turn program synthesis. *arXiv* **2022**, arXiv:2203.13474.
99. Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; Casas, D.d.L.; Hendricks, L.A.; Welbl, J.; Clark, A.; et al. Training compute-optimal large language models. *arXiv* **2022**, arXiv:2203.15556.
100. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. PaLM: Scaling language modeling with pathways. *J. Mach. Learn. Res.* **2023**, *24*, 1–113.
101. Black, S.; Biderman, S.; Hallahan, E.; Anthony, Q.; Gao, L.; Golding, L.; He, H.; Leahy, C.; McDonell, K.; Phang, J.; et al. GPT-NeoX-20B: An open-source autoregressive language model. *arXiv* **2022**, arXiv:2204.06745.
102. Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X.V.; et al. OPT: Open pre-trained transformer language models. *arXiv* **2022**, arXiv:2205.01068.
103. Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; et al. Solving quantitative reasoning problems with language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 3843–3857.
104. Soltan, S.; Ananthakrishnan, S.; FitzGerald, J.; Gupta, R.; Hamza, W.; Khan, H.; Peris, C.; Rawls, S.; Rosenbaum, A.; Rumshisky, A.; et al. AlexaTM 20B: Few-shot learning using a large-scale multilingual seq2seq model. *arXiv* **2022**, arXiv:2208.01448.
105. Le Scao, T.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A.S.; Yvon, F.; Gallé, M.; et al. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv* **2023**, arXiv:2211.05100.
106. Taylor, R.; Kardas, M.; Cucurull, G.; Scialom, T.; Hartshorn, A.; Saravia, E.; Poulton, A.; Kerkez, V.; Stojnic, R. Galactica: A large language model for science. *arXiv* **2022**, arXiv:2211.09085.
107. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
108. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. GPT-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
109. Ren, X.; Zhou, P.; Meng, X.; Huang, X.; Wang, Y.; Wang, W.; Li, P.; Zhang, X.; Podolskiy, A.; Arshinov, G.; et al. PanGu-Σ: Towards trillion parameter language model with sparse heterogeneous computing. *arXiv* **2023**, arXiv:2303.10845.
110. Wu, S.; Irsoy, O.; Lu, S.; Dabravolski, V.; Dredze, M.; Gehrmann, S.; Kambadur, P.; Rosenberg, D.; Mann, G. BloombergGPT: A large language model for finance. *arXiv* **2023**, arXiv:2303.17564.
111. Anil, R.; Dai, A.M.; Firat, O.; Johnson, M.; Lepikhin, D.; Passos, A.; Shakeri, S.; Taropa, E.; Bailey, P.; Chen, Z.; et al. PaLM 2 technical report. *arXiv* **2023**, arXiv:2305.10403.
112. Gunasekar, S.; Zhang, Y.; Aneja, J.; Mendes, C.C.T.; Del Giorno, A.; Gopi, S.; Javaheripi, M.; Kauffmann, P.; de Rosa, G.; Saarikivi, O.; et al. Textbooks are all you need. *arXiv* **2023**, arXiv:2306.11644.
113. Li, Y.; Bubeck, S.; Eldan, R.; Del Giorno, A.; Gunasekar, S.; Lee, Y.T. Textbooks are all you need II: Phi-1.5 technical report. *arXiv* **2023**, arXiv:2309.05463.

114. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. LLaMA 2: Open foundation and fine-tuned chat models. *arXiv* **2023**, arXiv:2307.09288.
115. Anthropic. Model Card and Evaluations for Claude Models. 2023. Available online: https://paperswithcode.com/paper/model-card-and-evaluations-for-claude-models (accessed on 23 September 2024).
116. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; Casas, D.d.l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv* **2023**, arXiv:2310.06825.
117. Almazrouei, E.; Alobeidli, H.; Alshamsi, A.; Cappelli, A.; Cojocaru, R.; Debbah, M.; Goffinet, É.; Hesslow, D.; Launay, J.; Malartic, Q.; et al. The Falcon series of open language models. *arXiv* **2023**, arXiv:2311.16867.
118. Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; Millican, K.; et al. Gemini: A family of highly capable multimodal models. *arXiv* **2023**, arXiv:2312.11805.
119. Javaheripi, M.; Bubeck, S.; Abdin, M.; Aneja, J.; Bubeck, S.; Mendes, C.C.T.; Chen, W.; Del Giorno, A.; Eldan, R.; Gopi, S.; et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*. 2023. Available online: https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/ (accessed on 23 September 2024).
120. Reid, M.; Savinov, N.; Teplyashin, D.; Lepikhin, D.; Lillicrap, T.; Alayrac, J.b.; Soricut, R.; Lazaridou, A.; Firat, O.; Schrittwieser, J.; et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv* **2024**, arXiv:2403.05530.
121. Mesnard, T.; Hardin, C.; Dadashi, R.; Bhupatiraju, S.; Pathak, S.; Sifre, L.; Rivière, M.; Kale, M.S.; Love, J.; Tafti, P.; et al. Gemma: Open models based on Gemini research and technology. *arXiv* **2024**, arXiv:2403.08295.
122. Anthropic. The Claude 3 Model Family: Opus, Sonnet, Haiku. 2024. Available online: https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf (accessed on 23 September 2024).
123. The Mosaic Research Team. Introducing DBRX: A New State-of-the-Art Open LLM. 2024. Available online: https://www.databricks.com/blog/introducing-dbrx-new-state-art-open-llm (accessed on 23 September 2024).
124. Jiang, A.Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D.S.; Casas, D.d.l.; Hanna, E.B.; Bressand, F.; et al. Mixtral of experts. *arXiv* **2024**, arXiv:2401.04088.
125. AI at Meta. Introducing Meta LLaMA 3: The Most Capable Openly Available LLM to Date. 2024. Available online: https://ai.meta.com/blog/meta-llama-3/ (accessed on 23 September 2024).
126. Abdin, M.; Jacobs, S.A.; Awan, A.A.; Aneja, J.; Awadallah, A.; Awadalla, H.; Bach, N.; Bahree, A.; Bakhtiari, A.; Behl, H.; et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv* **2024**, arXiv:2404.14219.
127. Chen, J.; Lin, H.; Han, X.; Sun, L. Benchmarking large language models in retrieval-augmented generation. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; Volume 38, pp. 17754–17762.
128. Salemi, A.; Zamani, H. Evaluating retrieval quality in retrieval-augmented generation. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington, DC, USA, 14–18 July 2024; pp. 2395–2400.
129. Wu, D.; Wang, X.; Qiao, Y.; Wang, Z.; Jiang, J.; Cui, S.; Wang, F. NetLLM: Adapting large language models for networking. *arXiv* **2024**, arXiv:2402.02338.
130. Zou, H.; Zhao, Q.; Bariah, L.; Bennis, M.; Debbah, M. Wireless multi-agent generative AI: From connected intelligence to collective intelligence. *arXiv* **2023**, arXiv:2307.02757.
131. Mongaillard, T.; Lasaulce, S.; Hicheur, O.; Zhang, C.; Bariah, L.; Varma, V.S.; Zou, H.; Zhao, Q.; Debbah, M. Large language models for power scheduling: A user-centric approach. *arXiv* **2024**, arXiv:2407.00476.
132. Liu, H.; Zhang, Z.; Wu, Q.; Zhang, Y. Large language model aided QoS prediction for service recommendation. *arXiv* **2024**, arXiv:2408.02223.
133. Desai, B.; Patel, K. Reinforcement learning-based load balancing with large language models and edge intelligence for dynamic cloud environments. *J. Innov. Technol.* **2023**, *6*, 1–13.
134. Chen, Y.; Li, R.; Zhao, Z.; Peng, C.; Wu, J.; Hossain, E.; Zhang, H. NetGPT: An AI-native network architecture for provisioning beyond personalized generative services. *IEEE Netw.* **2024**. [CrossRef]
135. Mondal, R.; Tang, A.; Beckett, R.; Millstein, T.; Varghese, G. What do LLMs need to synthesize correct router configurations? In Proceedings of the 22nd ACM Workshop on Hot Topics in Networks, Cambridge, MA, USA, 28–29 November 2023; pp. 189–195.
136. Lian, X.; Chen, Y.; Cheng, R.; Huang, J.; Thakkar, P.; Xu, T. Configuration validation with large language models. *arXiv* **2023**, arXiv:2310.09690.
137. Wang, C.; Scazzariello, M.; Farshin, A.; Kostic, D.; Chiesa, M. Making network configuration human friendly. *arXiv* **2023**, arXiv:2309.06342.
138. Dzeparoska, K.; Lin, J.; Tizghadam, A.; Leon-Garcia, A. LLM-based policy generation for intent-based management of applications. In Proceedings of the 2023 19th International Conference on Network and Service Management (CNSM), Niagara Falls, ON, Canada, 30 October–2 November 2023; pp. 1–7.
139. Zhang, T.; Feng, T.; Alam, S.; Dimitriadis, D.; Zhang, M.; Narayanan, S.S.; Avestimehr, S. GPT-FL: Generative pre-trained model-assisted federated learning. *arXiv* **2023**, arXiv:2306.02210.
140. Jiang, J.; Liu, X.; Fan, C. Low-parameter federated learning with large language models. *arXiv* **2023**, arXiv:2307.13896.
141. Meng, R.; Mirchev, M.; Böhme, M.; Roychoudhury, A. Large language model guided protocol fuzzing. In Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 26 February–1 March 2024.

142. Mekrache, A.; Ksentini, A. LLM-enabled intent-driven service configuration for next generation networks. In Proceedings of the 2024 IEEE 10th International Conference on Network Softwarization (NetSoft), Saint Louis, MO, USA, 24–28 June 2024; pp. 253–257.
143. Ifland, B.; Duani, E.; Krief, R.; Ohana, M.; Zilberman, A.; Murillo, A.; Manor, O.; Lavi, O.; Kenji, H.; Shabtai, A.; et al. GeNet: A multimodal LLM-based co-pilot for network topology and configuration. *arXiv* **2024**, arXiv:2407.08249.
144. Jeong, E.D.; Kim, H.G.; Nam, S.; Yoo, J.H.; Hong, J.W.K. S-Witch: Switch configuration assistant with LLM and prompt engineering. In Proceedings of the NOMS 2024–2024 IEEE Network Operations and Management Symposium, Seoul, Republic of Korea, 6–10 May 2024; pp. 1–7.
145. Mekrache, A.; Ksentini, A.; Verikoukis, C. Intent-based management of next-generation networks: An LLM-centric approach. *IEEE Netw.* **2024**, *38*, 29–36. [CrossRef]
146. Fuad, A.; Ahmed, A.H.; Riegler, M.A.; Čičić, T. An intent-based networks framework based on large language models. In Proceedings of the 2024 IEEE 10th International Conference on Network Softwarization (NetSoft), Saint Louis, MO, USA, 24–28 June 2024; pp. 7–12.
147. Louis, A. NetBERT: A Pre-Trained Language Representation Model for Computer Networking. Ph.D. Thesis, Cisco Systems, San Jose, CA, USA, 2020.
148. Setianto, F.; Tsani, E.; Sadiq, F.; Domalis, G.; Tsakalidis, D.; Kostakos, P. GPT-2C: A parser for honeypot logs using large pre-trained language models. In Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Kusadasi, Turkey, 6–9 November 2021; pp. 649–653.
149. Dietmüller, A.; Ray, S.; Jacob, R.; Vanbever, L. A new hope for network model generalization. In Proceedings of the 21st ACM Workshop on Hot Topics in Networks, Austin, TX, USA, 14–15 November 2022; pp. 152–159.
150. Qi, J.; Huang, S.; Luan, Z.; Yang, S.; Fung, C.; Yang, H.; Qian, D.; Shang, J.; Xiao, Z.; Wu, Z. LogGPT: Exploring ChatGPT for log-based anomaly detection. In Proceedings of the 2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Melbourne, Australia, 17–21 December 2023; pp. 273–280.
151. Lee, Y.; Kim, J.; Kang, P. LAnoBERT: System log anomaly detection based on BERT masked language model. *Appl. Soft Comput.* **2023**, *146*, 110689. [CrossRef]
152. Wang, J.; Zhang, L.; Yang, Y.; Zhuang, Z.; Qi, Q.; Sun, H.; Lu, L.; Feng, J.; Liao, J. Network meets ChatGPT: Intent autonomous management, control and operation. *J. Commun. Inf. Netw.* **2023**, *8*, 239–255. [CrossRef]
153. Gupta, P.; Kumar, H.; Kar, D.; Bhukar, K.; Aggarwal, P.; Mohapatra, P. Learning representations on logs for AIOps. In Proceedings of the 2023 IEEE 16th International Conference on Cloud Computing (CLOUD), Chicago, IL, USA, 2–8 July 2023; pp. 155–166.
154. Han, X.; Yuan, S.; Trabelsi, M. LogGPT: Log anomaly detection via GPT. In Proceedings of the 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, 15–18 December 2023; pp. 1117–1122.
155. Szabó, Z.; Bilicki, V. A new approach to web application security: Utilizing GPT language models for source code inspection. *Future Internet* **2023**, *15*, 326. [CrossRef]
156. Piovesan, N.; De Domenico, A.; Ayed, F. Telecom language models: Must they be large? *arXiv* **2024**, arXiv:2403.04666.
157. Jiang, Z.; Liu, J.; Chen, Z.; Li, Y.; Huang, J.; Huo, Y.; He, P.; Gu, J.; Lyu, M.R. LILAC: Log parsing using LLMs with adaptive parsing cache. *Proc. ACM Softw. Eng.* **2024**, *1*, 137–160. [CrossRef]
158. Kan, K.B.; Mun, H.; Cao, G.; Lee, Y. Mobile-LLaMA: Instruction fine-tuning open-source LLM for network analysis in 5G networks. *IEEE Netw.* **2024**, *38*, 76–83. [CrossRef]
159. Wong, H.; Luo, T. Man-in-the-middle attacks on MQTT-based IoT using BERT based adversarial message generation. In Proceedings of the KDD 2020 AIoT Workshop, Washington, DC, USA, 15–24 August 2020; Volume 8.
160. Ranade, P.; Piplai, A.; Joshi, A.; Finin, T. CyBERT: Contextualized embeddings for the cybersecurity domain. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; pp. 3334–3342.
161. Rahali, A.; Akhloufi, M.A. MalBERT: Using transformers for cybersecurity and malicious software detection. *arXiv* **2021**, arXiv:2103.03806.
162. Aghaei, E.; Niu, X.; Shadid, W.; Al-Shaer, E. SecureBERT: A domain-specific language model for cybersecurity. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Virtually, 17–19 October 2022; pp. 39–56.
163. Demırcı, D.; şahın, N.; şirlancis, M.; Acarturk, C. Static malware detection using stacked BiLSTM and GPT-2. *IEEE Access* **2022**, *10*, 58488–58502. [CrossRef]
164. Le, F.; Wertheimer, D.; Calo, S.; Nahum, E. NorBERT: Network representations through BERT for network analysis & management. In Proceedings of the 2022 30th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Nice, France, 18–20 October 2022; pp. 25–32.
165. Kholgh, D.K.; Kostakos, P. PAC-GPT: A novel approach to generating synthetic network traffic with GPT-3. *IEEE Access* **2023**, *11*, 114936–114951. [CrossRef]
166. Hamadanian, P.; Arzani, B.; Fouladi, S.; Kakarla, S.K.R.; Fonseca, R.; Billor, D.; Cheema, A.; Nkposong, E.; Chandra, R. A holistic view of AI-driven network incident management. In Proceedings of the 22nd ACM Workshop on Hot Topics in Networks, Cambridge, MA, USA, 28–29 November 2023; pp. 180–188.
167. Guo, H.; Yang, J.; Liu, J.; Yang, L.; Chai, L.; Bai, J.; Peng, J.; Hu, X.; Chen, C.; Zhang, D.; et al. Owl: A large language model for IT operations. *arXiv* **2023**, arXiv:2309.09298.

168. Mani, S.K.; Zhou, Y.; Hsieh, K.; Segarra, S.; Eberl, T.; Azulai, E.; Frizler, I.; Chandra, R.; Kandula, S. Enhancing network management using code generated by large language models. In Proceedings of the 22nd ACM Workshop on Hot Topics in Networks, Cambridge, MA, USA, 28–29 November 2023; pp. 196–204.
169. Bariah, L.; Zou, H.; Zhao, Q.; Mouhouche, B.; Bader, F.; Debbah, M. Understanding Telecom language through large language models. In Proceedings of the GLOBECOM 2023—2023 IEEE Global Communications Conference, Lumpur, Malaysia, 4–8 December 2023; pp. 6542–6547.
170. Tann, W.; Liu, Y.; Sim, J.H.; Seah, C.M.; Chang, E.C. Using large language models for cybersecurity Capture-the-Flag challenges and certification questions. *arXiv* **2023**, arXiv:2308.10443.
171. Kaheh, M.; Kholgh, D.K.; Kostakos, P. Cyber Sentinel: Exploring conversational agents in streamlining security tasks with GPT-4. *arXiv* **2023**, arXiv:2309.16422.
172. Moskal, S.; Laney, S.; Hemberg, E.; O'Reilly, U.M. LLMs killed the script kiddie: How agents supported by large language models change the landscape of network threat testing. *arXiv* **2023**, arXiv:2310.06936.
173. Piggott, B.; Patil, S.; Feng, G.; Odat, I.; Mukherjee, R.; Dharmalingam, B.; Liu, A. Net-GPT: A LLM-empowered man-in-the-middle chatbot for unmanned aerial vehicle. In Proceedings of the 2023 IEEE/ACM Symposium on Edge Computing (SEC), Wilmington, DE, USA, 6–9 December 2023; pp. 287–293.
174. Sarabi, A.; Yin, T.; Liu, M. An LLM-based framework for fingerprinting internet-connected devices. In Proceedings of the 2023 ACM on Internet Measurement Conference, Nice, France, 24–26 October 2023; pp. 478–484.
175. Ali, T.; Kostakos, P. HuntGPT: Integrating machine learning-based anomaly detection and explainable AI with large language models (LLMs). *arXiv* **2023**, arXiv:2309.16021.
176. Zhang, X.; Chen, T.; Wu, J.; Yu, Q. Intelligent network threat detection engine based on open source GPT-2 model. In Proceedings of the 2023 International Conference on Computer Science and Automation Technology (CSAT), Shanghai, China, 6–8 October 2023; pp. 392–397.
177. Wang, T.; Xie, X.; Zhang, L.; Wang, C.; Zhang, L.; Cui, Y. ShieldGPT: An LLM-based framework for DDoS mitigation. In Proceedings of the 8th Asia-Pacific Workshop on Networking, Sydney, Australia, 3–4 August 2024; pp. 108–114.
178. Ferrag, M.A.; Ndhlovu, M.; Tihanyi, N.; Cordeiro, L.C.; Debbah, M.; Lestable, T.; Thandi, N.S. Revolutionizing cyber threat detection with large language models: A privacy-preserving BERT-based lightweight model for IoT/IIoT devices. *IEEE Access* **2024**, *12*, 23733–23750. [CrossRef]
179. Habib, M.A.; Rivera, P.E.I.; Ozcan, Y.; Elsayed, M.; Bavand, M.; Gaigalas, R.; Erol-Kantarci, M. LLM-based intent processing and network optimization using attention-based hierarchical reinforcement learning. *arXiv* **2024**, arXiv:2406.06059.
180. Li, Q.; Zhang, Y.; Jia, Z.; Hu, Y.; Zhang, L.; Zhang, J.; Xu, Y.; Cui, Y.; Guo, Z.; Zhang, X. DoLLM: How large language models understanding network flow data to detect Carpet Bombing DDoS. *arXiv* **2024**, arXiv:2405.07638.
181. Manias, D.M.; Chouman, A.; Shami, A. Towards Intent-Based Network Management: Large Language Models for Intent Extraction in 5G Core Networks. In Proceedings of the 2024 20th International Conference on the Design of Reliable Communication Networks (DRCN), Montreal, QC, Canada, 6–9 May 2024; pp. 1–6. [CrossRef]
182. Chiang, M.; Low, S.H.; Calderbank, A.R.; Doyle, J.C. Layering as optimization decomposition: A mathematical theory of network architectures. *Proc. IEEE* **2007**, *95*, 255–312. [CrossRef]
183. Tang, A.; Wang, J.; Low, S.H.; Chiang, M. Equilibrium of heterogeneous congestion control: Existence and uniqueness. *IEEE/ACM Trans. Netw.* **2007**, *15*, 824–837. [CrossRef]
184. Low, S.H.; Lapsley, D.E. Optimization flow control. I. Basic algorithm and convergence. *IEEE/ACM Trans. Netw.* **1999**, *7*, 861–874. [CrossRef]
185. Jain, R.; Ramakrishnan, K.; Chiu, D.M. Congestion avoidance in computer networks with a connectionless network layer. *arXiv* **1998**, arXiv:cs/9809094.
186. Chiu, D.M.; Kadansky, M.; Provino, J.; Wesley, J.; Bischof, H.; Zhu, H. A congestion control algorithm for tree-based reliable multicast protocols. In Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, New York, NY, USA, 23–27 June 2002; Volume 3, pp. 1209–1217.
187. Chiu, D.M.; Jain, R. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Comput. Networks ISDN Syst.* **1989**, *17*, 1–14. [CrossRef]
188. Tan, C.W. The value of cooperation: From AIMD to flipped classroom teaching. *ACM SIGMETRICS Perform. Eval. Rev.* **2022**, *49*, 8–13. [CrossRef]
189. Wei, D.X.; Jin, C.; Low, S.H.; Hegde, S. FAST TCP: Motivation, architecture, algorithms, performance. *IEEE/ACM Trans. Netw.* **2006**, *14*, 1246–1259. [CrossRef]
190. Low, S.H.; Peterson, L.L.; Wang, L. Understanding TCP Vegas: A duality model. *J. ACM* **2002**, *49*, 207–235. [CrossRef]
191. Shorten, R.; Wirth, F.; Leith, D. A positive systems model of TCP-like congestion control: Asymptotic results. *IEEE/ACM Trans. Netw.* **2006**, *14*, 616–629. [CrossRef]
192. Winstein, K.; Balakrishnan, H. TCP ex Machina: Computer-generated congestion control. *ACM SIGCOMM Comput. Commun. Rev.* **2013**, *43*, 123–134. [CrossRef]
193. Izhikevich, L.; Enghardt, R.; Huang, T.-Y.; Teixeira, R. A global perspective on the past, present, and future of video streaming over Starlink. *arXiv* **2024**, arXiv:2409.09846.

194. Floyd, S. HighSpeed TCP for Large Congestion Windows. 2003. Available online: https://www.rfc-editor.org/rfc/rfc3649.html (accessed on 23 September 2024).
195. Chen, S.; Tan, C.W.; Zhai, X.; Poor, H.V. OpenRANet: Neuralized spectrum access by joint subcarrier and power allocation with optimization-based deep learning. *arXiv* **2024**, arXiv:2409.12964.
196. Tan, C.W.; Guo, S.; Wong, M.F.; Hang, C.N. Copilot for Xcode: Exploring AI-assisted programming by prompting cloud-based large language models. *arXiv* **2023**, arXiv:2307.14349.
197. Wong, M.F.; Guo, S.; Hang, C.N.; Ho, S.W.; Tan, C.W. Natural language generation and understanding of big code for AI-assisted programming: A review. *Entropy* **2023**, *25*, 888. [CrossRef] [PubMed]
198. Sheng, Y.; Cao, S.; Li, D.; Zhu, B.; Li, Z.; Zhuo, D.; Gonzalez, J.E.; Stoica, I. Fairness in serving large language models. In Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24), Santa Clara, CA, USA, 10–12 July 2024; pp. 965–988.
199. Jain, R.K.; Chiu, D.M.W.; Hawe, W.R. A quantitative measure of fairness and discrimination. *East. Res. Lab. Digit. Equip. Corp. Hudson, MA* **1984**, *21*, 1.
200. Chiu, D.M. Some observations on fairness of bandwidth sharing. In Proceedings of the ISCC 2000. Fifth IEEE Symposium on Computers and Communications, Antibes-Juan Les Pins, France, 3–6 July 2000; pp. 125–131.
201. Chiu, D.M.; Tam, A.S. Network fairness for heterogeneous applications. In Proceedings of the ACM SIGCOMM ASIA Workshop, Beijing, China, 12–14 April 2005.
202. Xu, Y.; Wang, Y.; Lui, J.C.; Chiu, D.M. Balancing throughput and fairness for TCP flows in multihop ad-hoc networks. In Proceedings of the 2007 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, Limassol, Cyprus, 16–20 April 2007; pp. 1–10.
203. Chiu, D.M.; Tam, A.S.W. Fairness of traffic controls for inelastic flows in the Internet. *Comput. Netw.* **2007**, *51*, 2938–2957. [CrossRef]
204. Fang, J.; He, Y.; Yu, F.R.; Li, J.; Leung, V.C. Large language models (LLMs) inference offloading and resource allocation in cloud-edge networks: An active inference approach. In Proceedings of the 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall), Hong Kong, China, 10–13 October 2023; pp. 1–5.
205. Bambhaniya, A.; Raj, R.; Jeong, G.; Kundu, S.; Srinivasan, S.; Elavazhagan, M.; Kumar, M.; Krishna, T. Demystifying platform requirements for diverse LLM inference use cases. *arXiv* **2024**, arXiv:2406.01698.
206. Zhang, J.; Vahidian, S.; Kuo, M.; Li, C.; Zhang, R.; Yu, T.; Wang, G.; Chen, Y. Towards building the FederatedGPT: Federated instruction tuning. In Proceedings of the ICASSP 2024–2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic Korea, 14–19 April 2024; pp. 6915–6919.
207. Cerf, V.G. Thoughts on AI interoperability. *Commun. ACM* **2024**, *67*, 5. [CrossRef]
208. Hadi, M.U.; Al Tashi, Q.; Shah, A.; Qureshi, R.; Muneer, A.; Irfan, M.; Zafar, A.; Shaikh, M.B.; Akhtar, N.; Wu, J.; et al. Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Prepr.* **2024**. [CrossRef]
209. Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R.; Amatriain, X.; Gao, J. Large language models: A survey. *arXiv* **2024**, arXiv:2402.06196.
210. AlZu'bi, S.; Mughaid, A.; Quiam, F.; Hendawi, S. Exploring the capabilities and limitations of ChatGPT and alternative big language models. In Proceedings of the Artificial Intelligence and Applications, Corfu, Greece, 22–24 February 2024; Volume 2, pp. 28–37.