Mobile-LLaMA: Instruction Fine-Tuning Open-Source LLM for Network Analysis in 5G Networks

Khen Bo Kan, Hyunsu Mun, Guohong Cao, and Youngseok Lee

Abstract

In the evolving landscape of 5G networks, Network Data Analytics Function (NWDAF) emerges as a key component, interacting with core network elements to enhance data collection, model training, and analytical outcomes. Language Models (LLMs), with their state-of-the-art capabilities in natural language processing, have been successful in numerous fields. In particular, LLMs enhanced through instruction fine-tuning have demonstrated their effectiveness by employing sets of instructions to precisely tailor the model's responses and behavior. However, it requires collecting a large pool of high-quality training data regarding the precise domain knowledge and the corresponding programming codes. In this paper, we present an open-source mobile network-specialized LLM, Mobile-LLaMA, which is an instruction-fine-tuned variant of the LLaMA 2 13B model. We build Mobile-LLaMA by instruction fine-tuning LLaMA 2 13B with our own network analysis data collected from publicly available, real-world 5G network datasets, and expanded its capabilities through a self-instruct framework utilizing OpenAI's pre-trained models (PMs). Mobile-LLaMA has three main functions: packet analysis, IP routing analysis, and performance analysis, enabling it to provide network analysis and contribute to the automation and artificial intelligence (AI) required for 5G network management and data analysis. Our evaluation demonstrates Mobile-LLaMA's proficiency in network analysis code generation, achieving a score of 247 out of 300 , surpassing GPT-3.5's score of 209.

Introduction

As many countries around the world have initiated 3GPP-compliant 5G services, we are now facing technical challenges beyond 5G wireless networks. To deploy various services of 5G networks such as enhanced mobile broadband (eMBB), ultra-reliable low latency communications (URLLC), and massive machine-type communications (mMTC), effectively managing the complexity of 5G network requires integrating network automation and AI for efficient data collection and analysis.

For 5G mobile network management, the 3rd Generation Partnership Project (3GPP) [1] has defined NWDAF, which collects, monitors, and analyzes data from the core network, generating insights into end-user performance and experience. It incorporates interfaces from the service-based architecture to collect and analyze data from user equipment (UE), network functions (NF), and operations, administration, and maintenance (OAM) systems from the 5G Core, Cloud, and Edge networks. NWDAF also implements machine intelligence for a centralized, predictive analytics platform for 5G core networks. 3GPP lists the following AI-ML analytics use cases for 5G using NWDAF [2]: load-level computation and prediction for a network slice instance; load analytics information and prediction for an application or UE group; network load performance computation and prediction; UE abnormal behavior or anomaly detection; UE mobility monitoring and prediction; UE communication pattern prediction; QoS reporting and prediction.

Large Language Models (LLMs), one of the most recent advanced generative AI technologies, have presented the possibility of intelligent edge computing services for beyond 5G wireless networks. As generative pre-trained transformers (GPT) have demonstrated knowledge understanding and code generation capabilities in natural language, many applications in various domains have applied LLM to their services. Besides playing a key role in providing chatbots or question-answering services for general or domain-specific knowledge, LLM is also being used to assist experts in their jobs. Although GPT models have demonstrated advanced performance, many private institutions or companies, including mobile carriers, cannot use commercial LLM services with their private user and network data. However, open-source LLMs such as Meta's LLaMA have enabled institutions that cannot provide their data to proprietary services to build their private LLM services.

In order to leverage LLM for intelligent network management of beyond 5G wireless networks, in this paper, we present a private training architecture for mobile network instructions and fine-tuned mobile network LLM for NWDAF. For

Digital Object Identifier: 10.1109/MNET.2024.3421306
Date of Current Version: 16 September 2024
Date of Publication: 3 July 2024

Khen Bo Kan, Hyunsu Mun, and Youngseok Lee (corresponding author) are with the Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, Republic of Korea; Guohong Cao is with the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802 USA.

0890-8044/24©2024IEEE

5G mobile network management and operation, we need to collect and analyze network, traffic and routing data, which requires knowledge of 5G mobile networks, network management and operation, and anomaly detection insight. Through mobile network-aware LLM, network management and operation could be performed more efficiently and automatically. However, mobile carriers should protect their user, network and performance data while employing LLM for intelligent network management. Therefore, LLM services for their networks should be based on open-source LLMs. In this paper, we demonstrate how we build a mobile network-specific LLM service with an open-source LLM, LLaMA 2 13B. We use 15,000 instruction-output sets, with 10,000 for IP routing analysis, 2,000 dedicated to mobile packet analysis, and the remaining 3,000 sets focused on performance analysis tasks in `Python`.

The main contributions of this paper include:

- Mobile network-aware fine-tuned LLM: we present Mobile-LLaMA,[1] a mobile network-specialized fine-tuned LLM, offering three key functions: IP routing analysis for managing IP flows and BGP paths, packet analysis for `pcap` file processing, and performance analysis for evaluating network performance.
- Comprehensive benchmark for network analysis tasks: we introduce a robust evaluation benchmark for Mobile-LLaMA. The benchmark assesses the model's proficiency in `Python` code generation for network analysis, focusing on its ability to effectively utilize specialized network libraries.

Mobile-LLaMA provides network management intelligence for NWDAF in 5G networks and establishes a new paradigm for LLMs suitable for 5G networks.

Related Work

Recent developments in natural language processing have significantly advanced LLMs, with generative pre-trained transformers like OpenAI's GPT models setting new standards in language understanding and generation. Instruction fine-tuning [3], a technique involving further training of LLM with instruction-output pairs to refine their responses to human input, has proven effective in enhancing its capabilities and control in the accuracy, fluency, and coherence of natural language responses.

In the network domain, LLMs are utilized for simpler tasks such as setup processes and synthetic data generation. LLMs can help network engineers simplify complex setup processes for tools like `Core Emulator` and `Wireshark`. They can also generate synthetic network traffic data, providing a cost-effective solution for training intrusion detection systems and addressing the scarcity of realistic cybersecurity datasets. However, their application in generating code scripts for network analysis remains unexplored.

Retrieval-Augmented Generation (RAG) [4] offers an alternative to fine-tuning by combining pre-trained transformers with dense vector indexes from external knowledge sources, enabling models to retrieve relevant information during content generation. While RAG improves language quality, it doesn't enhance LLM's coding capabilities for syntax and logical structures required in network analysis, which often involves specialized libraries for code generation.

Large Language Models (LLMs), one of the most recent advanced generative AI technologies, have presented the possibility of intelligent edge computing services for beyond 5G wireless networks.

Mobile-LLaMA for NWDAF

As shown in Fig. 1, NWDAF interacts with core network functions to collect data, train models, and provide analytic results. It collects information from various elements involved in access management, session management, and user plane operations and uses trained models to provide statistical or predictive information. User-related data can be used for model training and analytic results. Network policies and conditions can optimize network management and service delivery. Network Exposure Function (NEF) enables third-party applications to access network capabilities and functions through standardized interfaces.

NWDAF architecture consists of four primary components: data collection, Model Training Logical Function (MTLF), model store, and Analytics Logic Function (AnLF). Data collection involves retrieving and aggregating data from NFs and OAM systems. MTLF processes this data to train AI models. The trained models are then stored in model store. Finally, AnLF accesses the relevant model from model store and analyzes incoming data from data collection to generate analytical results.

Though NWDAF has been standardized in 5G, its development and implementation are still in early stages. Therefore, we propose a method to implement and enhance NWDAF in 5G using LLM. Based on open-source LLM, LLaMA 2 13B, we introduce Mobile-LLaMA, which utilizes aggregated data including network traffic patterns, user connectivity statistics, performance metrics, infrastructure details, and financial information, to evaluate network expansion options, optimize resource allocation, and guide infrastructure development. Network operators with NWDAF can leverage Moibile-LLaMA to perform IP routing analysis and traffic monitoring, to detect anomalies and configuration errors in 5G.

Overview of Mobile-LLaMA for 5G Networks

We suggest Mobile-LLaMA as a key component of NWDAF in 5G. Though NWDAF includes network analytics, LLM-supported functions will transform the operator-centric network management into the automated NWDAF. For the open-source LLM, we utilize Meta's LLaMA, which we instruction fine-tune to analyze 5G and backhaul data. We envision Mobile-LLaMA providing IP routing, packet, and performance analysis functions within 5G network NWDAF architecture in Fig. 1.

In NWDAF, MTLF is responsible for instruction fine-tuning Mobile-LLaMA by providing a set of instruction-output pairs. Each instruction serves as a prompt with the expected output that LLM is designed to generate. Once Mobile-LLaMA model is trained, it is saved in model store for future retrieval. When NF or AF within the network requires analytic insights, AnLF leverages the capabilities of Mobile-LLaMA model to analyze

[1] Mobile-LLaMA's codes and dataset are publicly available at the following GitHub repository https://github.com/DNLab2024/Mobile-LLaMA.

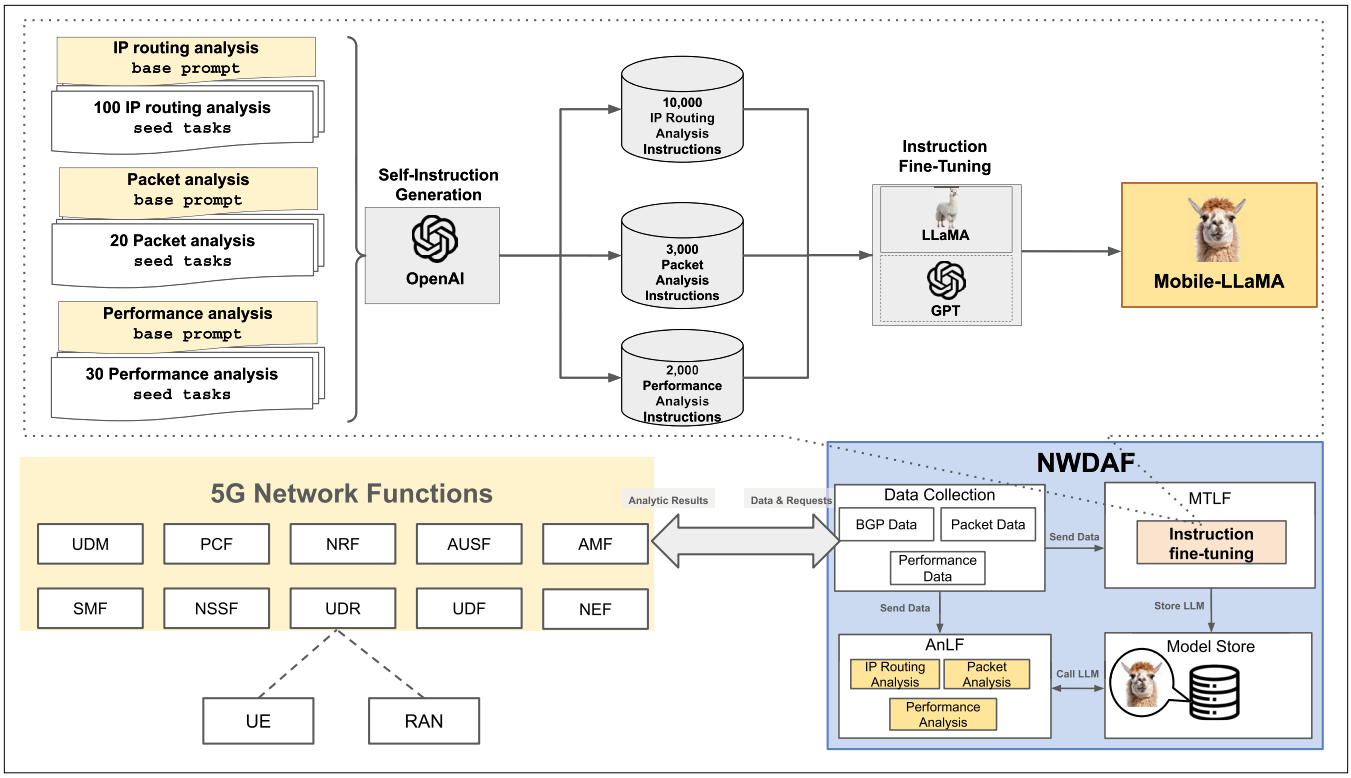

FIGURE 1. Architecture of NWDAF featuring Mobile-LLaMA for 5G network analytics. 5G network functions: Access and Mobility Management Function (AMF), Session Management Function (SMF), User Plane Function (UPF), Network Exposure Function (NEF), Unified Data Management (UDM), Policy Control Function (PCF), Network Repository Function (NRF), Authentication Server Function (AUSF), Network Slice Selection Function (NSSF), User Data Repository (UDR), and User Data Function (UDF). NWDAF components: Data Collection, Model Training Logical Function (MTLF), Model Store, and Analytics Logic Function (AnLF).

network data sent by data collection component. By providing the model with relevant queries, Mobile-LLaMA generates code scripts and AnLF extracts analytics results to support network optimization.

Functions of Mobile-LLaMA

Mobile-LLaMA integrates three main functions for mobile network analysis: IP routing analysis, packet analysis, and performance analysis as shown in Table 1. Although the base LLaMA models can produce basic `Python` scripts, it struggles with generating code for network analysis. Through fine-tuning with scripts that integrate specialized libraries, Mobile-LLaMA enhances its script generation for greater reliability and fewer errors. IP routing analysis, utilizing `PyBGPStream`, efficiently manages IP packet flow and monitors BGP path changes for network reliability. Packet analysis function leverages `Scapy` library to process packet capture files in `pcap` for parsing and analyzing IP packets. Performance analysis function uses `Pandas` for data manipulation and evaluation of network performance metrics, including capacity enhancements, cost-effectiveness, and overall network quality. The capabilities of Mobile-LLaMA across its three main functions are demonstrated in Fig. 2, and complete prompts, code, and code outputs for each demonstration is available in our GitHub repository.

1) IP Routing Analysis Function: IP routing analysis function utilizes IP routing messages to manage the flow of IP packets through the 5G backhaul network, ensuring efficient packet delivery to their intended destinations. Though it is possible for Mobile-LLaMA to handle intra- or inter-routing protocol messages, we focus on BGP routing table and update messages, because BGP routing table and messages are publicly available. We fine-tuned Mobile-LLaMA with BGP analysis instructions and the corresponding `Python` codes using `PyBGPStream`, allowing it to gather and analyze both real-time and historical BGP routing update messages. Mobile-LLaMA generates `Python` scripts that perform a detailed analysis of BGP routing tables, identifying advertised paths, detecting route changes, anomalies, and potential configuration issues or deviations in BGP protocols. The routing analysis function enables network operators to promptly address BGP route anomalies, thereby preventing network inefficiencies, outages, or disruptions, enhancing the utility and responsiveness of network management.

2) Packet Analysis Function: Packet analysis function is responsible for analyzing the IP packet data collected from 5G network elements, including the RAN, transport, and core network. Upon receiving instructions, Mobile-LLaMA generates `Python` scripts capable of executing a range of packet analysis tasks. Initially, it employs `Scapy` for parsing packet records from `pcap` files and uses `Pandas` to organize the data into `DataFrames`, preparing it for further analysis. Mobile-LLaMA then produces scripts to perform several key tasks: calculating round-trip times (RTT) for TCP connections or UDP flows, analyzing packet sizes to identify network inefficiencies

or anomalies, assessing vital network performance indicators such as throughput, latency, and packet loss rate, and conducting Quality of Service (QoS) assessments. The QoS evaluations include examining jitter, which indicates the variability in the time delay of packet arrivals, and assessing service delays.

3) Performance Analysis Function: Performance analysis function encompasses the evaluation of 5G network performance, utilizing structured data in `CSv` and `DataFrame` formats to provide network operators with analytical insights for infrastructure planning and optimization decisions. Given performance data and instructions, Mobile-LLaMA generates `Python` scripts using `Pandas` for data processing and analysis. The scripts execute precise calculations and handle data manipulations, including managing null values, selecting appropriate values for calculations, and identifying outliers. The analysis conducted by the generated scripts covers a variety of tasks: comparing LTE and 5G spectrums to calculate peruser capacity increases, evaluating investment cost efficiency across various geotypes (e.g. urban, suburban, rural), and assessing network quality by analyzing jitter, the Channel Quality Indicator (CQI), and detecting anomalies in UE traffic.

Implementation of Mobile-LLaMA

To implement Mobile-LLaMA for 5G NWDAF, we leverage fine-tuning capability of an open-source LLM, LLaMA 2 13B. We need mobile network training data and instructions to enable the 5G network analysis functions. We use publicly available datasets such as IP packets in `pcap`, performance data in `csv`, as well as IP routing in `mrt` formats, and expand the training data by applying self-instruct [6] method to GPT.

1) Training Data: For instruction fine-tuning, we collect publicly available 5G network datasets, which align with the operational functions of Mobile-LLaMA within NWDAF. For IP routing analysis function, we utilize public BGP routing tables in `mrt` format. The tables consist of both raw and formatted BGP routing data, archived by the RouteViews and RIPE projects. For packet analysis function, we utilize a dataset provided by Deutsche Telekom's GitHub repository [7] providing `pcap` data of 5G network traffic. For performance analysis function, we train Mobile-LLaMA with two

Recent developments in natural language processing have significantly advanced LLMs, with generative pre-trained transformers like OpenAI's GPT models setting new standards in language understanding and generation.

distinct datasets both presented in `csv` format. The first [8] contains detailed time series data, covering UE network performance metrics like throughput, latency, and CQI in LTE/5G networks. The second dataset [9] provides data to various 5G infrastructure strategies, focusing on capacity, coverage, and cost. Utilizing real-world datasets, we have developed a collection of high-quality `Python` code scripts specifically for 5G data analysis instructions. The primary goal is to use these scripts as a training foundation for the pre-trained Language Model - LLaMA. Each script is crafted to effectively demonstrate the use of essential data analysis libraries, including `PyBGPStream` for IP routing analysis, `Scapy` for packet analysis, `Pandas` for data manipulation, `Matplotlib` for visualization. This strategy guarantees that our fine-tuned model is not only proficient in handling the complexities of 5G network data but also capable of performing a wide range of tasks

Function of Mobile-LLaMA	Analysis Tasks	Library	Manual Instruction Tasks	Self-Instruct Generated Instructions
IP routing analysis	BGP route anomalies BGP path changes	`PyBGPStream`	100	10000
Packet analysis	Parsing IP packets Data structuring RTT calculation Packet size examination Performance assessment QoS assessment	`Scapy`	20	2000
Performance analysis	Per-user capacity Enhancement 5G investment cost analysis Network quality evaluation UE traffic anomaly detection Jitter & CQI benchmarking	`Pandas` `Matplotlib`	30	3000

TABLE 1. Analysis tasks in NWDAF of 5G network with mobile-LLaMA and training data summary.

FIGURE 2. Demonstrations of Mobile-LLaMA's capabilities in a) IP routing analysis, b) packet analysis, and c) performance analysis.

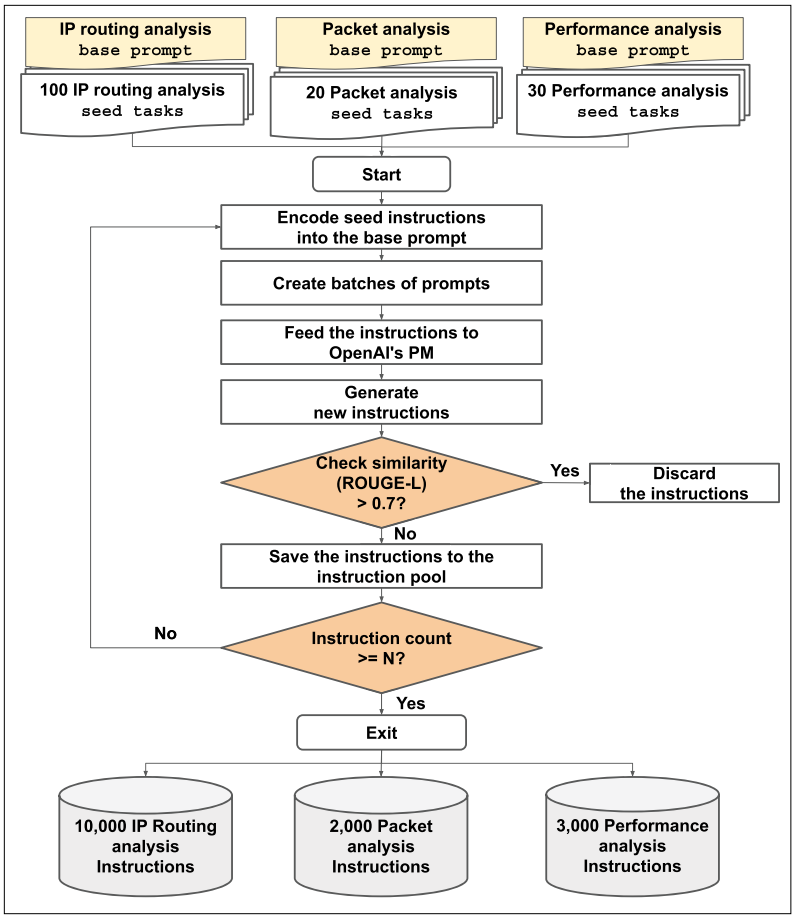

FIGURE 3. High-level overview of the self-instruct generation process.

across the three main functions of Mobile-LLaMA in 5G network data analysis.

Crafting a substantial set of high-quality training instructions for 5G network analysis is a meticulous and resource-intensive process that typically involves initial writing, multiple rounds of revisions, and expert validation. The complexity of 5G network analysis requires deep technical knowledge for the creation of precise instructions, which can be a challenge when trying to gather a large pool of such specialized experts. For the development of Mobile-LLaMA training dataset, we adopt self-instruct method that leverages OpenAI's pre-trained models (PMs), significantly diversifying and expanding our pool of initial instruction tasks as demonstrated in Fig. 3. Self-instruct framework begins with two essential elements: manually-written `seed tasks` and the `base prompt`. We utilized `Python` code scripts for 5G data analysis as the basis to create manual instruction `seed tasks`, designed to guide the PM in generating new instructions. However, self-instruct inherits limitations of PMs, including errors in generated code when using niche libraries, such as `Scapy` and `PyBGPStream`. To address this, we incorporate guidelines on the library use into the `base prompt`, specifying correct syntax, parameters and class instantiation. The process begins by loading the `base prompt` and encoding multiple `seed tasks` into the prompt. It then proceeds to create batches of prompts, each batch comprising a combination of the `base prompt` and a subset of seed instructions. The combined prompts are fed into the OpenAI's PM, which produces potential new instructions. We guide the model in generating new instructions that align with the context and format of the seed instructions. To ensure the uniqueness and relevance of the generated instructions, we employ a similarity check using the ROUGE-L score. If the similarity score of a new instruction with any of the existing instructions exceeds a threshold of 70%, the instruction is discarded to avoid redundancy. The process continues iteratively, generating batches of instructions until the desired number of unique instructions is achieved. With self-instruct framework, we have expanded our dataset from initial 150 manual `seed tasks` across three main functions of Mobile-LLaMA: 100 for IP routing analysis, 20 for packet analysis, and 30 for performance analysis, to a comprehensive set of 15,000 instructions distributed as follows: 10,000 instructions for IP routing analysis, 2,000 for packet analysis, and 3,000 for performance analysis, as detailed in Table 1. Examples of the `base prompt`, `seed tasks`, and self-instruct-generated instructions are shown in our GitHub repository.[2]

2) Instruction Fine-Tuning LLaMA for 5G Analytics: We select LLaMA 2 13B as our base model for 5G network analysis due to its optimal balance between performance and usability, outperforming LLaMA 2 7B and being more practical than the resource-intensive LLaMA 2 70B. For instruction fine-tuning, we prepare the dataset in the structured prompt that describes a task to the base model with expected response. In total, we have 15,000 instruction-output samples in which we reserved 12,000 for training and 3,000 for validation. The prompts are tokenized into sequences with max token length of 2048. However, fine-tuning LLaMA 2 model [10], especially with a large number of parameters, presents challenges due to the computational resources, particularly in terms of VRAM on GPUs of commodity machines. To overcome the problem, we adopt Low-Rank Adaptation (LoRA) [11] for parameter-efficient fine-tuning. By introducing new trainable parameters in the form of low-rank matrices, this approach reduces the VRAM requirements, making it feasible to fine-tune large models on commodity hardware setups.

The model trains for 5,000 steps with a learning rate at 1×10^{-4} and a global batch size of 4 with a warm-up ratio of 0.05 . The process uses a cosine learning rate schedule for gradual adjustments, a gradient accumulation step count of 1 , and a maximum gradient norm of 0.3 for precise and incremental learning adjustments. The optimizer used is AdamW 32-bit. Training uses FP16 for computational efficiency and precision. We set LoRA hyperparameters with a rank of 64, α with a value of 16 , dropout rate of 0.1, and bias excluded to excluded to focus on the primary weight matrices. The fine-tuning process spans over 3 epochs, involves 52,428,800 trainable parameters, and has been completed in 10 hours on two NVIDIA RTX A6000 48GB GPUs.

Performance Evaluation

HumanEval [12] is a standard benchmark for evaluating the code generation abilities of LLMs. HumanEval comprises of a collection of hand-written programming problems, each consisting of

[2] https://github.com/DNLab2024/Mobile-LLaMA/tree/main/images/selfInstructFigs

a function signature, docstring, body, and multiple unit tests. The evaluation process involves presenting these tasks to the models to generate code solutions, which are then compared against reference solutions to measure accuracy using the binary evaluation system - pass/fail. The primary metric used for evaluation is the `pass@1` metric, which quantifies the percentage of generated solutions that match the reference solutions on the first attempt. HumanEval is utilized in performance evaluation of many LLMs for code generation, including Codex, GPT-J, GPT-Neo, GPT-NeoX-20B, and CodeParrot [13].

However, HumanEval focuses on evaluating LLMs through function-level or statement-level code generation tasks, such as generating a function to return a specific value. In contrast, the main functions of Mobile-LLaMA are designed for more complex tasks involving collecting BGP routing tables and messages, processing `pcap`, and analysing 5G network data. Moreover, HumanEval's binary evaluation system oversimplifies the assessment process into a pass or fail, and does not capture the different levels of correctness in the model outputs. Recognizing the model's ability to partially solve problems is valuable for observing common patterns, errors, default behaviors, and identifying potential improvements. Given Mobile-LLaMA's focus on network analysis tasks, a more detailed evaluation that comprehends the complexity of network operations and captures varying degrees of correctness is essential.

Metric for Network Code Evaluation

To address the limitation of HumanEval, we develop a custom benchmark for Mobile-LLaMA. Our benchmark is structured into 30 distinct tasks, distributed evenly across three main categories. IP routing analysis category involves detecting anomalies and configuration errors in network paths using `PyBGPStream` library. Packet analysis category emphasizes the proper use of the `Scapy` library for the examination of network traffic in `pcap` format. Performance analysis category assesses 5G network infrastructure and metrics, with tasks such as 5G QoS parameters analysis and UE traffic volume. Each category includes 10 specific tasks designed to evaluate its respective aspects of 5G network analysis. Our two invited experts evaluate the code generated by LLM models based on guidelines and practices taken from the official GitHub repositories for PyBGPStream [14] and Scapy [15]. Our evaluation process targets the first iteration of code output. Each output starts with a maximum potential score of 10 points, with deductions for necessary corrections as outlined in Table 2. Examples of error types can be viewed in our GitHub repository. Our point deduction system ensures the code follows end-user prompt requirements, provides insights into models' performance in executing network-related tasks, and offers a more granular approach to the evaluation process. Our evaluation result shows that Mobile-LLaMA boasts an impressive performance, scoring 247 out of possible 300 points. Fig. 4 illustrates the evaluation of Mobile-LLaMA performance compared to LLaMA 2 7B, 13B, 70B and GPT-3.5 (Turbo 16K) across all three main evaluation categories.

Error type	Explanation	Deduction Points
Syntax errors	Code contains syntax errors that prevent it from running	2
Error and exception handling	Lack of or inadequate handling of potential errors or exceptions	2
Incomplete implementation	Code only partially addresses the task and is not fully implemented	2-3
Logical errors	Code runs but results in incorrect outcomes due to logical mistakes	4-6
Inaccuracy in data handling	Issues in correctly processing or interpreting data, leading to inaccurate or unreliable results	5-7
Incorrect library usage	Improper or inefficient use of libraries relevant to the task for network analysis.	7

TABLE 2. Point deduction criteria for code evaluation.

Code Assessment for IP Routing Analysis

In IP routing analysis code assessment, as shown in Fig. 4a, Mobile-LLaMA demonstrated remarkable proficiency, securing an impressive score of 97 out of a possible 100 points. In comparison, GPT scored significantly lower, with 49 points. The base LLaMA 2 models (7B, 13B, and 70B) scored 0 points across all three model sizes, due to their lack of familiarity with the `PyBGPStream` library. The results marked a substantial performance gap between Mobile-LLaMA and GPT-3.5 in this category. Recurring issues with the GPT-3.5 include mistakes in date and time formatting and incorrect utilization of the "filter" and "collector" parameters. The correct application of these parameters is crucial for collecting both real-time and historical BGP routing update messages, as well as for analyzing BGP routing tables, thereby enabling a clear understanding of the advertised BGP paths. As a result, it completely failed to effectively handle time-series tasks and to accurately manage access to live stream data sources. On the other hand, Mobile-LLaMA demonstrates its capability by mostly generating error-free codes. The only deducted points for Mobile-LLaMA occasional mistake of considering live stream data access as a historical data point.

Code Assessment for Packet Analysis

In the packet analysis code assessment, Mobile-LLaMA demonstrated results similar to GPT-3.5, scoring 80 out of a possible 100 points, as shown in Fig. 4b. The base LLaMA 2 models predominantly use the outdated `Pcapy` library, resulting in frequent inaccuracies in their generated code. Even when prompted to use more current libraries like `Scapy`, their outputs often appear as hallucinations, consequently receiving a score 0 points. GPT-3.5, while achieving a slightly higher score of 82 points, maintained a marginal lead over Mobile-LLaMA. The lead is attributed to more effective handling of error/exception scenarios, particularly in managing various network layers like IP, TCP, and Raw.

Code Assessment for Performance Analysis

Fig. 4c shows performance analysis code assessment. Mobile-LLaMA scored 70, marking a modest improvement over the base LLaMA 2 models 7B, 13B, 70B, which achieved scores of 27,58 , and 62 respectively. GPT-3.5 performed slightly better, with a score of 78. The modest

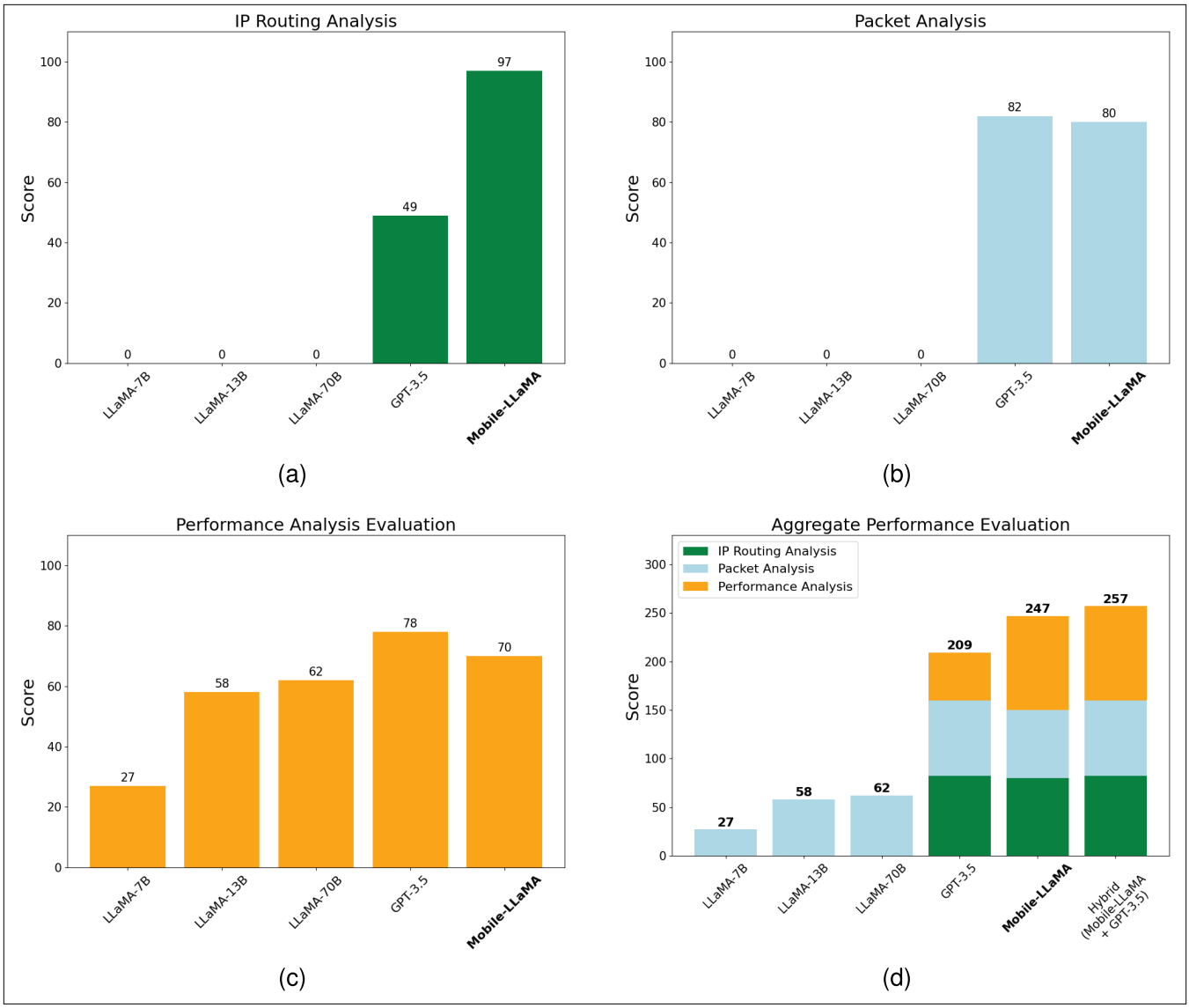

FIGURE 4. Comparison of performance evaluation results: a) IP routing analysis, b) Packet analysis, c) Performance analysis, d) Aggregate performance evaluation scores - LLaMA 2 Models, GPT-3.5 (Turbo 16K), Mobile-LLaMA, Hybrid (GPT-3.5 + Mobile-LLaMA).

progression in Mobile-LLaMA's performance is due to the wide range of tasks in performance analysis, each requiring different analytical methods. Both Mobile-LLaMA and GPT-3.5 faced challenges in generating correct code for complex plotting tasks involving grouping and categorization. Additionally, both models exhibited limitations in handling null or zero values in datasets, affecting the accuracy of the results. However, GPT-3.5 showed comparatively more effective data handling in these instances than Mobile-LLaMA.

As shown in Fig. 4d, the hybrid approach that combines GPT-3.5 for packet and performance analysis and Mobile-LLaMA for IP routing analysis achieves the highest evaluation score of 257 .

Discussion

Our instruction fine-tuning has enabled Mobile-LLaMA to produce quality code for IP routing analysis. One example[3] is continuous monitoring of routing table updates in real time to identify anomalies that may indicate misconfigurations or security threats. Utilizing Mobile-LLaMA for IP routing analysis can potentially reduce dependency on third-party tools and lower operational costs of network management. However, Mobile-LLaMA's capability in packet and performance analysis functions is still slightly behind GPT-3.5. We attribute this to a smaller number of `seed tasks` of 20 and 30 respectively, compared to 100 `seed tasks` for IP routing analysis. We observe that simply increasing the instruction count on existing `seed tasks` does not improve performance without collecting a more diverse range of analysis cases. However, diversifying `seed tasks` is challenging due to limited real-world data availability, restricted by privacy concerns.

When deciding between open-source private and commercial public LLMs, mobile carriers have to consider their specific needs for cost-efficiency, privacy, and performance. Deploying an open-source private LLM by following our framework requires a substantial initial investment in infrastructure. On-premise server equipped with eight NVIDIA A100 GPU can exceed $200,000.

[3] https://github.com/DNLab2024/Mobile-LLaMA/tree/main/evaluation/BestPractice

However, an open-source private LLM guarantees privacy and ensures sensitive data is not exposed to third-party entities, and offers transparency and customizability. Conversely, while instruction fine-tuning commercial public LLMs may offer higher performance, the long-term financial cost becomes considerably higher with increase in request volume due to API usage and cloud-based server infrastructure. Commercial public models charge based on the prompts and generated output, with GPT-4 costing $30.00 for input and $60.00 for output per million tokens, and the hourly cost for a single cloud-based GPU server like AWS A100 being approximately $32.70. However, commercial public models allow for simpler infrastructure setup and fewer specialized personnel for LLM maintenance.

Future work on network-specific LLMs focuses on broadening Mobile-LLaMA's functions. Mobile-LLaMA can be fine-tuned with telcom domain knowledge, business terms and conversation records between customers and service agents to enable summarization, intent recognition, and topic generation. Equipping these capabilities allows Mobile-LLaMA to identify and act on customer needs. The framework of Mobile-LLaMA can be implemented with other open-source LLMs to compare performance, resource usage and effectiveness. Federated instruction tuning (FedIT) [16] addresses data scarcity by decentralizing the training process across multiple client devices. Each device holds a local instruction dataset and independently updates small, trainable adapters within the pre-trained model. Through federated learning, clients enhance the model without sharing sensitive data, updating LLM locally, and returning only the modified adapters for aggregation. The application of FedIT within the Mobile-LLaMA framework can provide valuable insights into its efficacy compared to traditional centralized fine-tuning methods.

Conclusion

In this paper, we introduce Mobile-LLaMA within NWDAF framework for 5G networks, extending the functionalities of standard NWDAF with a instruction-finetuned LLM for comprehensive network analysis. Mobile-LLaMA's capabilities in mobile network analysis are highlighted through its three main functions: IP routing analysis, 5G packet analysis and performance analysis functions. We also establish a comprehensive benchmark for evaluating Mobile-LLaMA's performance across these functions. Mobile-LLaMA achieved a notable score of 247 out of 300 in our comprehensive evaluation, surpassing base LLaMA 2 models across all categories, and outperforming GPT-3.5 in IP routing analysis. The result highlights the effectiveness of our instruction fine-tuning approach for network analysis tasks.

Acknowledgment

This work was supported by the Research Fund of Chungnam National University.

Future work on network-specific LLMs focuses on broadening Mobile-LLaMA's functions. Mobile-LLaMA can be finetuned with telcom domain knowledge, business terms and conversation records between customers and service agents to enable summarization, intent recognition, and topic generation.

References

[1] *3GPPArchitecture Enhancements for 5G System (5GS) to Support Network Data Analytics Services*, Standard (TS) 23.288, version 18.1.0, 3rd Generation Partnership Project (3GPP), Technical Specification, Mar. 2023. Accessed: Sep. 10, 2023. [Online]. Available: https://www.3gpp.org/DynaReport/23288.htm

[2] M. A. Garcia-Martin, M. Gramaglia, and P. Serrano, "Network automation and data analytics in 3GPP 5G systems," *IEEE Netw.*, early access, Oct. 6, 2023, doi: 10.1109/MNET.2023.3321524.

[3] H. W. Chung et al., "Scaling instruction-finetuned language models," 2022, *arXiv:2210.11416*.

[4] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," 2020, *arXiv:2005.11401*.

[5] S. A. Marques, D. Z. Rodriguez, and R. L. Rosa, "Use of ChatGPT as configuration support tool and network analysis," in *Proc. Int. Conf. Softw., Telecom Munications Comput. Netw. (SoftCOM)*, Sep. 2023, pp. 1–6, doi: 10.23919/SoftCOM58365.2023.10271595.

[6] Y. Wang et al., "Self-instruct: Aligning language models with SelfGenerated instructions," 2023, *arXiv:2212.10560*.

[7] Deutsche Telekom. *GitHub Repository: 5G-Trace-Visualizer*. Accessed: Nov. 10, 2023. [Online]. Available: https://github.com/telekom/5g-trace-visualizer#plotting-scripts

[8] T. Tsourdinis et al., "UE network traffic time-series (applications, throughput, latency, CQI) in LTE/5G networks," *IEEE Dataport*, 2022, doi 10.21227/4ars-fs38. Accessed: Nov. 10, 2023.

[9] E. Oughton et al., "Data for: Assessing the capacity, coverage and cost of 5G infrastructure strategies: Evidence from The Netherlands," *Mendeley Data, V1*, vol. 37, pp. 50–69, 2019, doi: 10.17632/n656zp56cy.1.

[10] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.

[11] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," 2021, *arXiv:2106.09685*.

[12] M. Chen et al, "Evaluating large language models trained on code," 2021, *arXiv:2107.03374*.

[13] F. F. Xu et al., "A systematic evaluation of large language models of code," in *Proc. 6th ACM SIGPLAN Int. Symp. Mach. Program.*, Jun. 2022, pp. 1–10, doi: 10.1145/3520312.3534862.

[14] CAIDA. *GitHub Repository: PyBGPStream*. Accessed: Jun. 14, 2023. [Online]. Available: https://github.com/CAIDA/pybgpstream

[15] Philippe Biondi. *GitHub Repository: Scapy*. Accessed: Jun. 14, 2023. [Online]. Available: https://github.com/secdev/scapy

[16] J. Zhang et al., "Towards building the federated GPT: Federated instruction tuning," 2023, *arXiv:2305.05644*.

Biographies

Khen Bo Kan (hyonbokan@o.cnu.ac.kr) is currently pursuing the master's degree in computer science and engineering with Chungnam National University, South Korea.

Hyunsu Mun (munhyunsu@cnu.ac.kr) received the Ph.D. degree in computer science and engineering from Chungnam National University, South Korea.

Guohong Cao (Fellow, IEEE) (gxc27@psu.edu) is currently a Professor with the Department of Computer Science and Engineering, The Pennsylvania State University, USA.

Youngseok Lee (lee@cnu.ac.kr) is currently a Professor of computer science and engineering, Chungnam National University, South Korea.