INTELLIGENT NETWORK MANAGEMENT RADIO ACCESS NETWORKS USING FINE-TUNED LARGE LANGUAGE MODELS WITH MIXTURE-OF-AGENT ARCHITECTURE AND KNOWLEDGE GRAPH RETRIEVAL

THESIS PROPOSAL

Submitted as one of the requirements to obtain
Magister of Informatics

By:

MUHAMMAD AMRI
001202407013

FACULTY OF COMPUTER SCIENCE
MASTER OF INFORMATICS STUDY PROGRAM
CIKARANG
AUGUST, 2025

ABSTRACT

The increasing complexity of radio access networks (RAN) in modern telecommunications, driven by multi-vendor environments and evolving standards, has highlighted the limitations of current management systems in delivering intelligent, context-aware technical support. Existing tools often lack integration across diverse data sources and struggle to interpret detailed technical queries, resulting in inefficiencies and a higher risk of misconfiguration. This thesis addresses the gap by proposing a Mixture of Agent (MoA) Fine Tuned LLMs approach architecture that integrates with knowledge graph-based retrieval to support advanced question answering and technical decision support in RAN management. The research objectives are to design, implement, and evaluate the MoA-based system, focusing on its ability to unify heterogeneous data, interpret operator intent, and generate accurate, contextually relevant responses. The methodology involves developing the MoA architecture, curating and preprocessing RAN-specific datasets, and systematically assessing system performance using established LLM evaluation metrics such as accuracy, precision, recall, and relevance. Key findings demonstrate that the proposed approach improves the technical capabilities of RAN management systems, particularly in intelligent information retrieval and automated technical support. The results suggest that integrating LLMs through a multi-agent framework can address complex technical queries more effectively than traditional methods. The study concludes that the MoA-based architecture offers a promising direction for advancing intelligent, scalable solutions in next-generation RAN management. {Will Be Update after result}

Keyword: Radio Access Networks, Fine-tuned Large Language Models, Mixture of Agent Architecture, Knowledge Graph Retrieval, Intelligent Network Management

TABLE OF CONTENTS

Page

CHAPTER 1
INTRODUCTION

1.1. Background

The rapid evolution of telecommunications, particularly within radio access networks (RAN), has introduced increasing complexity in network configuration and management. As technologies have progressed from 2G through 5G, and with the emergence of Open RAN and the anticipation of 6G, operators face significant challenges in handling multi-vendor environments and diverse configuration parameters. The growing complexities of rule-based network management systems significantly increase the effort required for change management activities, as operators must manually update the rule-based management system periodically to accommodate evolving RAN requirements, leading to delays and higher risks of errors. These complexities directly demand more intelligent management solutions. This research addresses the need for advanced, automated approaches to RAN configuration and performance management by exploring the integration of large language models (LLMs) within intelligent network management systems. The primary goal is to develop and evaluate Mixture of Agent (MoA) Fine Tuned LLMs approach architecture for advanced question answering and knowledge graph-based retrieval-augmented generation, tailored to the unique data formats and operational requirements of RAN environments [1]. Through this work, the thesis aims to demonstrate how LLM-driven systems can enhance the efficiency and effectiveness of network management in modern telecommunications.

1.2. Research Problem

Despite the increasing adoption of automation and analytics in radio access network (RAN) management, current systems remain limited in their ability to provide intelligent, context-aware support for complex operational tasks. Existing tools often operate in isolation, lack integration across different data sources, and are unable to interpret detailed technical queries or deliver useful insights tailored to specific RAN environments. This separation leads to ongoing inefficiencies, a higher risk of misconfiguration, and challenges in maintaining optimal network performance, especially as RAN environments become more varied and dynamic.

The specific gap this thesis addresses is the absence of an adaptive, knowledge-driven system that uses large language models (LLMs) for advanced question answering and decision support in RAN management. There is a clear need for a solution that can bring together different data, understand operator intent, and generate precise, relevant responses to technical queries. Without such capabilities, network operators face ongoing difficulties in efficiently managing and troubleshooting increasingly complex RAN infrastructures.

To address this problem, the thesis proposes and evaluates Fine-Tuned Large Language Models based on Mixture-of-Agent Architecture with knowledge graph-based retrieval. The research will implement this method and systematically assess its effectiveness in unifying data sources, interpreting operator queries, and delivering technically accurate recommendations. By validating this approach, the study aims to show its potential to advance the technical capabilities of RAN management systems, especially in terms of intelligent network management information retrieval and automated technical support.

1.3. Research Objective

The objectives of this thesis are to clearly define and address the technical challenges involved in integrating large language models (LLMs) into radio access network (RAN) management. This research aims to design, implement, and evaluate a Mixture of Agent (MoA) architecture that combines fine-tuned LLMs with knowledge graph-based retrieval to support advanced question answering and technical decision support. By focusing on the technical aspects of system integration, information retrieval, and automated response generation, the study seeks to establish a robust framework for applying LLMs in the context of RAN environments. The research is guided by the following questions:

a. How can a Mixture of Agent (MoA) Fine Tuned LLMs approach architecture and knowledge graph-based retrieval, be designed to support advanced question answering and technical decision support in RAN environments?

b. How effective is the proposed approach in generating accurate and contextually relevant responses to technical queries, as measured by established LLM evaluation metrics such as accuracy, precision, recall, and relevance?

c. What is the impact of the MoA based system on the integration and interpretation of different RAN data sources for technical information retrieval, as validated through quantitative evaluation?

1.4. State of the Art

This research advances the state of the art in radio access network (RAN) management by demonstrating how large language models (LLMs), when integrated through a Mixture of Agent (MoA) architecture with knowledge graph-based retrieval, can address technical challenges unique to modern telecommunications environments. The study's significance lies in its potential to offer a novel approach for intelligent information retrieval and automated technical support within RAN systems, moving beyond traditional rule-based or siloed approaches.

Theoretically, this work contributes to the growing body of knowledge on applying LLMs to domain-specific, high-complexity technical fields. It offers insights into the design and evaluation of Mixture of Agent architectures that combine natural language understanding with structured data retrieval, providing a reference for future research in both telecommunications and AI-driven automation.

Practically, the findings may inform the development of more adaptive and scalable network management tools, enabling telecommunication engineers to respond more effectively to technical queries and configuration challenges. By validating the effectiveness of the proposed approach using established evaluation metrics, the study provides a foundation for future deployments of LLM-based solutions in real-world RAN environments, potentially improving network reliability, reducing manual workload, and accelerating troubleshooting processes.

1.5. Gap Analysis

Despite recent advances in automation and AI-driven solutions for radio access network (RAN) management, current approaches remain fragmented and limited in their ability to deliver comprehensive, context-aware technical support. Most existing systems rely on either generic large language models or isolated retrieval mechanisms, which struggle to interpret complex queries and unify diverse data sources. Multi-agent architectures and knowledge graph retrieval have shown promise in other domains, but their integration for RAN management is still nascent. There is a clear absence of a holistic framework that combines fine-tuned language models, collaborative agent reasoning, and structured knowledge retrieval to address the unique challenges of RAN environments. This gap results in persistent inefficiencies, limited adaptability, and a lack of robust decision support for network operators. The proposed research aims to bridge this gap by developing and evaluating an integrated Mixture of Agent architecture with knowledge graph-based retrieval, specifically tailored for intelligent network management in RAN scenarios.

CHAPTER 2
LITERATURE REVIEW

2.1. Radio Access Network

The evolution of Radio Access Networks represents one of the most significant technological transformations in modern telecommunications, characterized by increasing architectural complexity and operational challenges that demand intelligent management solutions. The progression from 2G through 5G networks has fundamentally altered the landscape of wireless communications, with each generation introducing new paradigms that compound management complexity [2].

Modern RANs are far denser and more complex than their predecessors. The shift to 5G (and future 6G) introduces denser cell deployment, massive MIMO antennas, ultra-low latency requirements, and new service types (e.g. IoT, edge cloud). For example, Lee et al. (2021) note that 6G RANs will demand "extremely high-performance interconnections" and highly diverse capabilities to support dynamic environments [3]. In Open RAN deployments, each network function may come from a different supplier, vastly expanding the number of possible configurations. In short, multi-vendor Open RAN environments require integration across diverse platforms and constant re-calibration of parameters, making manual management fragile and error prone.

2.2. Fine-tuned Large Language Model

Large Language Models (LLMs) like GPT, BERT, and their successors have shown astonishing general-purpose reasoning and generation abilities. However, telecom and RAN management tasks are highly specialized: they involve technical protocols, configuration formats, and multi-step procedures not found in general text. For instance, Zhou et al. (2024) note that the knowledge needed for network architecture, protocols, and standards improve LLM's performance for domain specific tasks, but the telecom specific dataset collection and filtering still required careful design and evaluation [4]. A study by Erak et al. (2025) similarly argues that while LLMs have great potential to automate network management, they must be adapted to deeply understand telecom nuances [5].

The two main adaptation strategies are fine-tuning and retrieval-augmentation (RAG). Fine-tuning further trains a pre-trained LLM on domain-specific data, aligning its internal knowledge to the target domain. RAG, by contrast, uses retrieval of external knowledge (often documents or structured data) to guide the model's output. Both approaches have pros and cons. Erak et al. explain that fine-tuning can adjust the model parameters for telecom data but is costly and static once tuned. It is hard to update without full retraining [5]. In contrast, RAG grounds the LLM's answers in up-to-date sources, making it easier to incorporate new knowledge, though it requires an external retrieval system [5]. Both methods aim to reduce hallucinations and improve precision. In practice, hybrid approaches often combine a fine-tuned LLM with RAG over a domain knowledge base (3GPP standards).

State-of-the-art techniques emphasize parameter-efficient fine-tuning. Training an LLM (with billions of parameters) from scratch or full fine-tune is very expensive. Techniques like Low-Rank Adaptation (LoRA) introduce a small number of trainable parameters into each layer of the transformer, achieving substantial speedups. As Parthasarathy et al. (2024) survey, methods such as LoRA, adapter layers, and even novel frameworks like Mixture-of-Agents (MoA) allow specialization with minimal compute [6]. For example, a recent telecom-specific LLM (TSLAM-Mini) fine-tuned a 3.8B-parameter base model on a curated dataset of 100,000 telecom examples using Quantized LoRA (QLoRA) [7]. This specialized model significantly outperformed generic LLMs on technical queries.

In summary, the literature on fine-tuned LLMs emphasizes adapting models to the technical telecom domain through either continued training or retrieval augmentation. Studies consistently find that domain fine-tuning significantly improves accuracy on network management queries, as long as high-quality, specialized datasets are used. Novel methods like parameter-efficient LoRA and hierarchical multi-agent LLMs (MoA) have been proposed to make fine-tuning more tractable. Overall, the consensus is that general LLMs must be carefully adapted via fine-tuning, RAG, or both to serve telecom use cases reliably.

2.3. Mixture Of Agent Architecture

Mixture-of-Agents (MoA) is an emerging LLM architecture in which multiple models (“agents”) collaborate to solve a task, drawing on ideas from mixture-of-experts and multi-agent systems. In a layered MoA design, each layer consists of several parallel LLM agents, and every agent in a layer can access the outputs of all agents in the previous layer [1]. This suggests that distributing reasoning across diverse models can be more powerful than relying on a single model instance.

The MoA pattern is closely related to broader concepts of multi-agent AI. In general, a multi-agent system consists of multiple autonomous AI agents that work together to solve problems. Modern LLM-based multi-agent frameworks explicitly assign different roles or skills to each agent. For example, one agent might specialize in network configuration syntax, another in performance analysis, and a third in protocol knowledge. These agents can exchange information or propose partial solutions iteratively. Guo, T et al. (2024) review LLM-based multi-agent approaches and note that this paradigm "leverages the collective intelligence and specialized skills of multiple agents" [8]. They highlight that agents can engage in multi-turn planning, debate, and decision-making, enabling complex tasks to be broken into subproblems.

Design-wise, MoA architectures borrow ideas from mixture-of-experts (MoE) neural networks but replace the expert gating with cooperative multi-agent consensus. Each agent is a full LLM, and instead of a learned router, the system may use a simple aggregator (e.g. averaging) or a learned controller to combine agent outputs [1]. This design allows each agent to focus on different facets of the query or bring varied knowledge. In some proposals, agents have distinct prompts or fine-tuning (so-called heterogeneous MoA), while in others they are identical copies working in parallel (homogeneous MoA). Regardless, the goal is that no single point of failure (as in a single LLM) exists, and that consensus among agents yields more robust answers. In network management, one can imagine assigning agents to different data modalities or vendor-specific knowledge bases.

In summary, mixture-of-agents architectures extend the single-LLM paradigm by using multiple interacting agents to tackle complex problems. This design promises improved accuracy (through ensemble effects) and flexibility (agents can be specialized or updated independently). In the context of intelligent RAN management, MoA is particularly attractive: the network is inherently multi-component and heterogeneous, matching the strengths of a multi-agent approach. Current research suggests that MoA can achieve state-of-the-art performance on language tasks, laying a foundation for its use in distributed AI systems like future automated network controllers.

2.4. Knowledge Graph Retrieval

Knowledge Graphs (KGs) are structured databases of entities and relationships that model domain knowledge in a machine-readable form. In the context of RAN management, a KG might capture network elements (cells, antennas, protocols) and their attributes or interconnections. Knowledge graph retrieval (KGR) refers to querying these structures to retrieve relevant facts or subgraphs that can inform decision-making or question-answering.

Recent work has focused on combining KGs with LLMs. One approach is retrieval-augmented generation (RAG), where an LLM is guided by external information retrieved from a KG. For example, Wang, S et al. (2025) observe that vanilla text-based RAG can introduce noise because it ignores structural relations. They propose “K-RAG,” which explicitly retrieves structured information from a knowledge graph to augment recommendations, aiming to reduce hallucinations [9]. In this model, instead of feeding raw documents to the LLM, high-precision triples from a KG are used as context. Such structured retrieval ensures that the LLM's answers can be grounded in verifiable facts and complex relations.

Other frameworks illustrate iterative KG-based retrieval. Jiang, P et al. (2024) propose a Retrieval and Structuring (RAS) method: given a query, the system retrieves relevant text, extracts triples to grow a query-specific KG, and then feeds the combined query graph into the LLM for final answer generation. They demonstrate that structuring knowledge allows the model to perform "systematic reasoning grounded in the assembled knowledge," rather than relying on latent patterns alone [10]. This kind of knowledge-grounded LLM is promising for RAN tasks: for example, a path computation question could be answered by traversing a graph of node links and capacity constraints, rather than hoping the LLM remembered network topology.

There is also increasing interest in LLM-driven KG construction and querying for telecom. Chen et al. (2025) propose a KG-RAG pipeline specifically for telecommunications QA. In their design, domain-specific LLMs extract telecom entities and relations into a KG; at query time, the most relevant subgraph is retrieved and fed into an LLM for answer generation. This approach ensures that the LLM's reply is conditioned on the retrieved knowledge and tailored to evolving telecom standards [11]. Earlier, Kumar et al. (2025) described a system where diverse data sources (mails, calendars, documents, activity logs, and other repositories) are unified into a comprehensive knowledge graph through LLM-assisted extraction. They show that querying this unified KG enables powerful analytics and decision support that were not possible with siloed data [12].

In summary, the literature shows that structured knowledge retrieval is emerging as a key technique for LLM-based QA in technical domains. By storing RAN and telecom domain information in a KG, systems can retrieve precise facts and relationships to inform LLM responses. Approaches like GraphRAG and KG-RAG demonstrate that embedding KG triples into the LLM's context can drastically improve answer accuracy and reduce hallucination. For intelligent network management, integrating LLMs with knowledge graphs promises contextualized, reliable information retrieval across heterogeneous data sources.

2.5. Intelligent Network Management

Intelligent network management refers to the trend of using automation, AI, and advanced analytics to operate and optimize networks with minimal human intervention. In the RAN context, this encompasses self-organizing networks (SON), intent-based networking, predictive maintenance, and closed-loop control. A key theme in the literature is that traditional manual or rule-based operations cannot scale to 5G/6G in complexity, so AI-driven automation is essential.

Self-organizing/self-healing networks are another focus. Studies in 5G management emphasize SON algorithms that handle tasks like self-configuration, self-optimization, and self-protection. A survey by Perumallapli (2015) on self-healing networks highlights the use of AI (especially ML and anomaly detection) for fault management. A self-healing network can "automatically identify, diagnose, [and] fix" problems without manual intervention, greatly reducing downtime [13]. For instance, if a cell site underperforms, an AI agent could detect unusual metrics, isolate the root cause (e.g. hardware failure or bad configuration), and apply corrective actions (reroute traffic or adjust parameters) automatically. This proactive approach contrasts with legacy O&M where faults are handled only after customer complaints.

Industry efforts mirror these trends. For example, the O-RAN Alliance has defined the RAN Intelligent Controller (RIC) as a platform for hosting AI/ML applications that optimize RAN functions in real-time. Telecom vendors also promote "self-driving networks" with AI agents that can learn from network data. These developments underscore that network management is shifting towards closed-loop and knowledge-driven processes. The theoretical literature sees this as an evolution of earlier autonomic networking research, now energized by powerful ML and LLM tools that can reason complex configurations and large datasets.

In practice, combining these AI-driven approaches has shown real benefits. Operators report that AI-based automation can speed up fault resolution and capacity provisioning. For instance, predictive models can forecast traffic hot spots so that capacity can be preemptively adjusted, rather than reacting after congestion occurs. Overall, the literature suggests a move toward cognitive network management – where understanding and decision-making are guided by data analytics and LLM reasoning, rather than solely by static rules or human expertise. This intelligence is seen as critical for managing the ever-more-complex RANs of today and tomorrow.

2.6. Related Works

Several recent studies have begun to integrate the above technologies in the context of network management. For example, Erak et al. (2024) demonstrate a fine-tuned RAG system tailored to telecommunications. Their framework, based on a Phi-2 small language model, is trained on 3GPP documents and uses retrieval of spec passages to answer network queries. They also created the TeleQuAD and TeleQnA datasets (telecom QA corpora) and built TeleRoBERTa, a BERT-based RAG QA model for telecom scenarios [5]. This line of work shows how LLMs can be specialized with domain data and combined with retrieval to handle technical questions.

On the knowledge graph front, Chen et al. (2025) develop a KG-RAG framework for telecom QA. Their system first extracts telecom entities and relations into a knowledge graph, then retrieves the relevant subgraph to ground the LLM's answers. They report that incorporating structured KG info significantly improves answer precision compared to plain LLM QA [11]. These works illustrate how RAN-specific knowledge graphs can be leveraged to inform LLM responses in network management.

Finally, the concept of multi-agent LLMs is beginning to influence related work. MoA as proposed by Wang, J et al. sets a precedent for layered agent collaboration [1]. While MoA itself has not yet been applied to RAN problems in literature, there are efforts to use multiple LLM agents for complex tasks (e.g. autonomous network planning via agent simulations). Guo, T. et al survey on LLM-based multi-agent systems highlights various prototypes where agents with different specialties coordinate to solve problems [8]. These approaches remain nascent but suggest that distributed LLM systems could manage different aspects of network data in tandem.

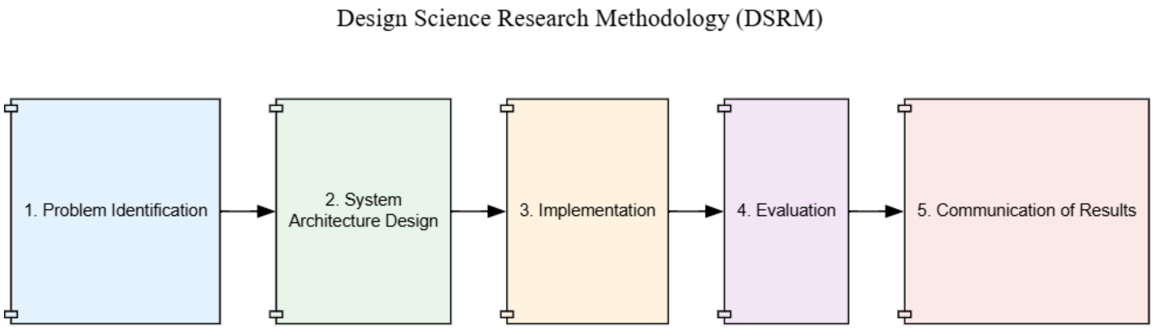
In summary, related research has been selected to converge LLM adaptation, multi-agent coordination, and knowledge retrieval for telecommunications. For example, specialized telecom LLMs (like TSLAM-Mini), telecom QA systems (TeleQuAD/RoBERTa), and KG-augmented LLMs (KG-RAG) have all demonstrated the potential of these components. However, a comprehensive MoA + RAG + KG solution for RAN remains a novel contribution. The present proposal distinguishes itself by aiming to combine fine-tuned LLM agents (MoA) with a knowledge-graph retrieval backend specifically for RAN management. To our knowledge, no prior work has simultaneously integrated all these elements for the RAN context.

CHAPTER 3
METHODOLOGY

3.1. Research Design

This research adopts a Design Science Research Methodology (DSRM) to address the technical challenges of intelligent network management in radio access networks (RAN) by integrating fine-tuned large language models (LLMs) within a Mixture of Agent (MoA) architecture, enhanced by knowledge graph retrieval. The Design Science Research Methodology paradigm is well-suited for this study, as it emphasizes the creation, implementation, and rigorous evaluation of innovative artifacts that solve identified problems in complex, real-world domains. In the context of RAN management, where the landscape is shaped by rapid technological evolution, multi-vendor environments, and diverse data sources, a design science methodology enables the systematic development and assessment of a novel comprehensive MoA + RAG + KG solution system.

The choice of a Design Science Research Methodology is motivated by the need to both advance theoretical understanding and deliver practical solutions [13]. Traditional empirical or purely analytical approaches may fall short in capturing the intricacies of integrating LLMs, Mixture of Agent approach, and structured knowledge retrieval within operational RAN environments. By contrast, Design Science Research Methodology allows for iterative prototyping, testing, and refinement of the proposed system, ensuring that the resulting architecture is both theoretically grounded and practically viable.


Design Science Research Methodology (DSRM)
1. Problem Identification
2. System Architecture Design
3. Implementation
4. Evaluation
5. Communication of Results

3.1 Design Science Research Methodology

Figure 3.1 shows the main steps of the research: identifying the problem, designing the system, building its modules, evaluating how well it works, and sharing the results. All these activities are part of the Methodology chapter and help ensure that the research process and reporting findings are closely connected. More details about the results and their meaning are given in the Results and Discussion chapters.

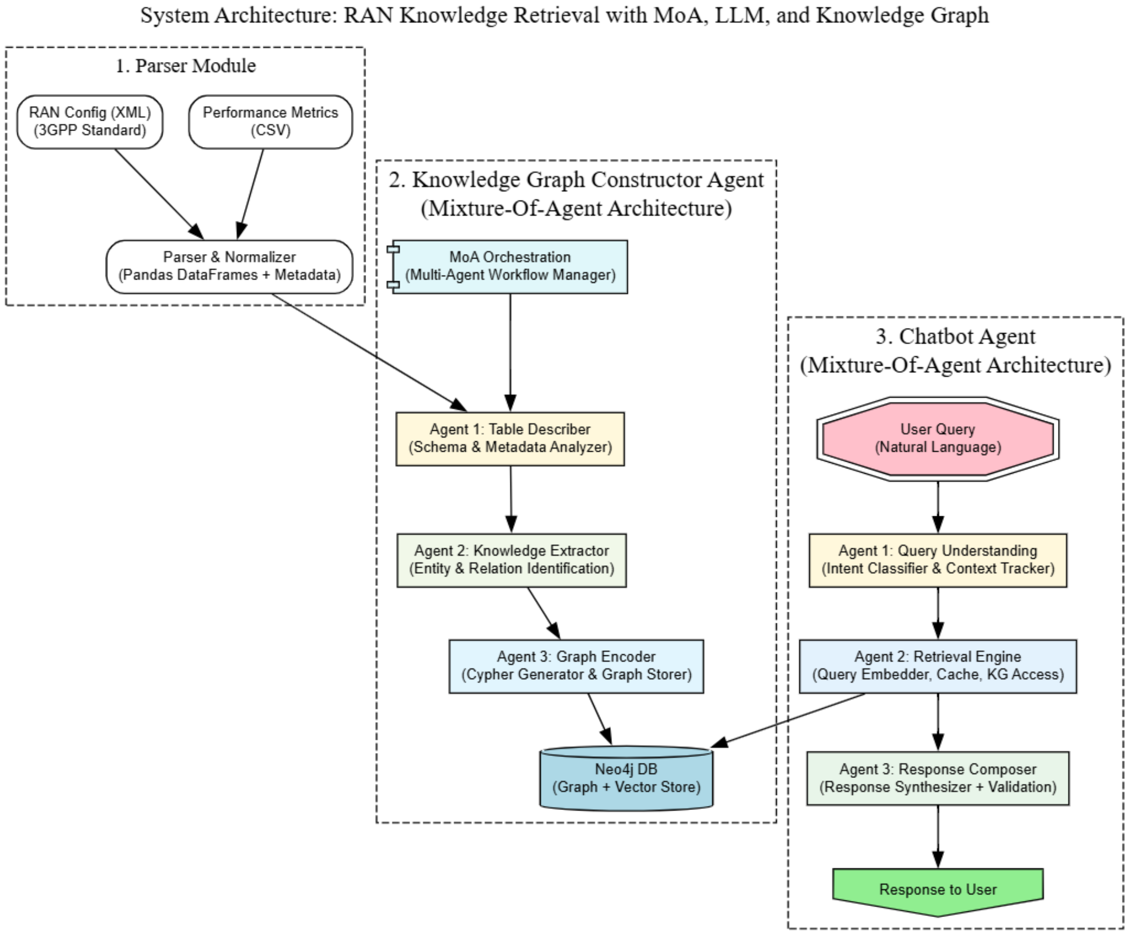
3.2. Problem Identification

The problem identification for this study centers on the escalating complexity and diversity of Radio Access Networks (RAN) in modern telecommunications, driven by rapid technological evolution, multi-vendor deployments, and the proliferation of diverse data sources. Traditional RAN management systems, which often rely on rule-based or siloed approaches, are increasingly inadequate for handling the nuanced, context-dependent technical challenges presented by 5G, Open RAN, and the anticipated 6G networks. These legacy systems struggle to integrate information across disparate platforms, interpret complex operator queries, and deliver timely, accurate technical support leading to inefficiencies, higher risks of misconfiguration, and suboptimal network performance.

The literature review highlights that while Large Language Models (LLMs) have demonstrated strong general reasoning capabilities, their effectiveness in telecom-specific tasks is limited without domain adaptation. Fine-tuning and retrieval-augmented generation (RAG) approaches have shown promise, but require high-quality, domain-specific datasets and robust integration with structured knowledge sources. Emerging architectures like Mixture-of-Agents (MoA) further suggest that distributing reasoning across specialized LLM agents can enhance accuracy and flexibility, yet their application to RAN management remains underexplored. Additionally, knowledge graph retrieval is recognized as a powerful method for grounding LLM outputs in verifiable, structured domain knowledge, but integrating this with multi-agent LLM systems for RAN management is still a novel area.

Therefore, the core problem addressed by this research is the lack of an adaptive, intelligent network management system that unifies fine-tuned LLMs, MoA architectures, and knowledge graph retrieval to provide advanced, context-aware technical support in complex RAN environments. This study seeks to design, implement, and evaluate such a system, aiming to bridge the gap between current limitations and the demands of next-generation network management.

3.3. System Architecture Design


System Architecture: RAN Knowledge Retrieval with MoA, LLM, and Knowledge Graph
1. Parser Module
RAN Config (XML)
(3GPP Standard)
Performance Metrics
(CSV)
Parser & Normalizer
(Pandas DataFrames + Metadata)
2. Knowledge Graph Constructor Agent
(Mixture-Of-Agent Architecture)
MoA Orchestration
(Multi-Agent Workflow Manager)
Agent 1: Table Describer
(Schema & Metadata Analyzer)
Agent 2: Knowledge Extractor
(Entity & Relation Identification)
Agent 3: Graph Encoder
(Cypher Generator & Graph Storer)
Neo4j DB
(Graph + Vector Store)
3. Chatbot Agent
(Mixture-Of-Agent Architecture)
User Query
(Natural Language)
Agent 1: Query Understanding
(Intent Classifier & Context Tracker)
Agent 2: Retrieval Engine
(Query Embedder, Cache, KG Access)
Agent 3: Response Composer
(Response Synthesizer + Validation)
Response to User

3.2 System Architecture Design

The architecture of the proposed intelligent network management system for Radio Access Networks (RAN) is conceived as a modular, pipeline-based framework that integrates advanced artificial intelligence and knowledge engineering techniques. As shown in Figure 3.2, the design and development of the proposed system are organized into three core functional components: the Parser Module, the Knowledge Graph Constructor Agent, and the Chatbot Agent. This architecture is designed to address the growing complexity and heterogeneity of modern RAN environments, where data originates from multiple sources, including configuration management systems and performance monitoring platforms. The system's architecture is intentionally structured to unify these diverse data streams, enable semantic enrichment, and support intelligent technical decision-making through seamless interaction between its core components.

The architecture begins with the ingestion of raw RAN data, which includes XML-based configuration files adhering to 3GPP standards and tabular performance management data in CSV format. The system is engineered to systematically transform and normalize this heterogeneous data, ensuring consistency and interoperability across the pipeline. Once standardized, the data undergoes semantic enrichment, where advanced processing techniques powered by Fine-tuned Large Language Models extract meaningful entities, relationships, and technical context. This enriched information is then structured into a knowledge graph, which serves as the backbone of the system. The knowledge graph is implemented using a platform such as Neo4j, chosen for its robust support of both graph-based and vector-based retrieval, enabling efficient storage, querying, and contextual reasoning.

A defining feature of the architecture is the integration of a Mixture of Agent Architecture, which distributes reasoning and problem-solving tasks across multiple specialized agents. These agents, each leveraging Fine-tuned Large Language Models, collaborate to interpret technical data, generate semantic annotations, and facilitate natural language interactions with users. The architecture is inherently scalable, allowing for the addition of new data sources, analytical modules, or agent specializations as network management needs evolve. This flexibility ensures that the system can adapt to the dynamic requirements of intelligent network management in RAN, supporting advanced question answering, technical support, and automated decision-making.

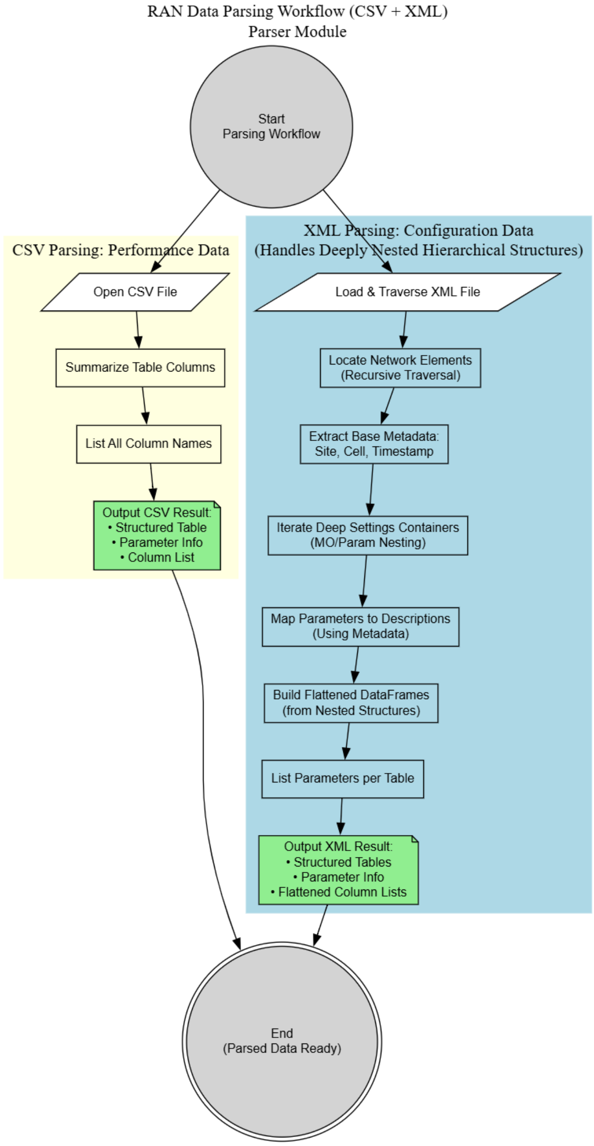
By orchestrating the flow of data from raw ingestion to intelligent query response, the architecture establishes a robust foundation for next generation RAN management. The design not only enhances operational efficiency and accuracy but also positions the system to accommodate future advancements in telecommunications and artificial intelligence. Through the seamless integration of Radio Access Networks, Fine-tuned Large Language Models with Mixture of Agent Architecture, Knowledge Graph Retrieval, and Intelligent Network Management, the architecture exemplifies a forward-looking approach to managing the complexities of modern wireless networks.

3.4. Module Implementation

The implementation of the intelligent network management system translates the architectural vision into a practical, modular solution. Each module is developed to fulfill a specific function within the overall pipeline, ensuring seamless integration and robust performance. The following subsections detail the design and practical realization of each core module: the Parser Module, the Knowledge Graph Constructor Agent, and the Chatbot Agent.

3.4.1. Parser Module

The Parser Module is a foundational component of the intelligent network management system, responsible for ingesting, transforming, and normalizing raw RAN data into a structured format suitable for downstream processing. Figure 3.3 illustrates the overall flow of the Parser Module, depicting how raw data from both 3GPP-compliant XML configuration files and CSV-based performance management tables is systematically processed. The figure highlights the sequential steps of data ingestion, parsing, normalization, and metadata extraction, culminating in the creation of unified Pandas DataFrames and associated metadata dictionaries. This visual representation underscores the module's role as the entry point for heterogeneous RAN data, ensuring that all subsequent modules receive consistent and well-structured inputs for further semantic enrichment and knowledge graph construction. Implemented in Python, this module leverages the Pandas library for efficient data manipulation and employs specialized parsing routines for both CSV and XML files, reflecting the diversity of data sources encountered in Radio Access Networks.



3.3 Parser Module Flow

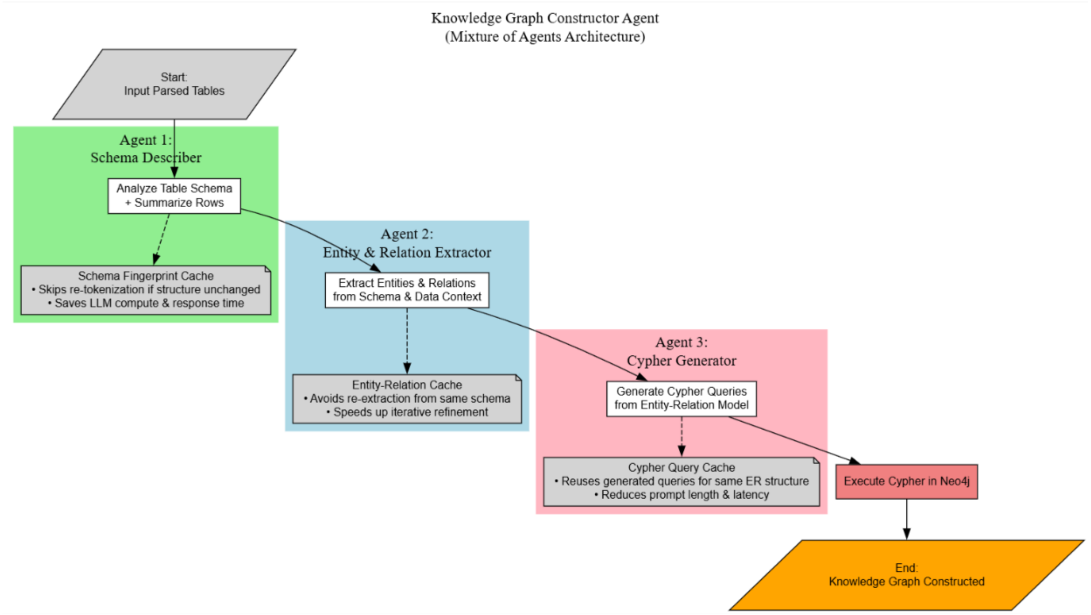
For tabular performance management data, the module uses Pandas to read CSV files, converting them into DataFrames while simultaneously extracting column names and generating metadata. This metadata includes a mapping of parameter names to placeholder descriptions, ensuring that each attribute is documented for later semantic enrichment and knowledge graph construction.

The XML parsing functionality is tailored to handle complex 3GPP-compliant RAN Configuration Management files, such as those produced by Ericsson systems. Utilizing Python’s xml.etree.ElementTree, the parser navigates the hierarchical structure of the XML, extracting relevant elements like SubNetwork, MeContext, ManagedElement, and VsDataContainer. It systematically collects key attributes such as area names, cell IDs, and nested configuration parameters while maintaining contextual information like date and network area. The parser is designed to handle multiple levels of nesting, ensuring that even deeply embedded configuration data is accurately captured.

Throughout the parsing process, the module constructs a dictionary of DataFrames, each representing a logical table (e.g., EnodeBInfo, or tables named after vsDataType values). It also builds two layers of metadata: a detailed mapping of parameters for each table and a simplified list of columns. This dual metadata approach supports both technical traceability and ease of integration with subsequent modules, such as the Knowledge Graph Constructor Agent. The extensible design of the Parser Module allows for the future inclusion of additional data formats or preprocessing routines, ensuring adaptability as RAN management requirements evolve.

3.4.2. Knowledge Graph Constructor Agent

The workflow and technical architecture of the Knowledge Graph Constructor Agent are illustrated in Figure 3.4. This central module enables Knowledge Graph Retrieval and Intelligent Network Management in Radio Access Networks. Built on a Mixture of Agent Architecture, each specialized agent is assigned a distinct functional responsibility to transform structured tabular data into a semantically rich knowledge graph.


Knowledge Graph Constructor Agent
(Mixture of Agents Architecture)
Start:
Input Parsed Tables
Agent 1:
Schema Describer
Analyze Table Schema
+ Summarize Rows
Schema Fingerprint Cache
• Skips re-tokenization if structure unchanged
• Saves LLM compute & response time
Agent 2:
Entity & Relation Extractor
Extract Entities & Relations
from Schema & Data Context
Entity-Relation Cache
• Avoids re-extraction from same schema
• Speeds up iterative refinement
Agent 3:
Cypher Generator
Generate Cypher Queries
from Entity-Relation Model
Cypher Query Cache
• Reuses generated queries for same ER structure
• Reduces prompt length & latency
Execute Cypher in Neo4j
End:
Knowledge Graph Constructed

3.4 Knowledge Graph Constructor Agent Flow

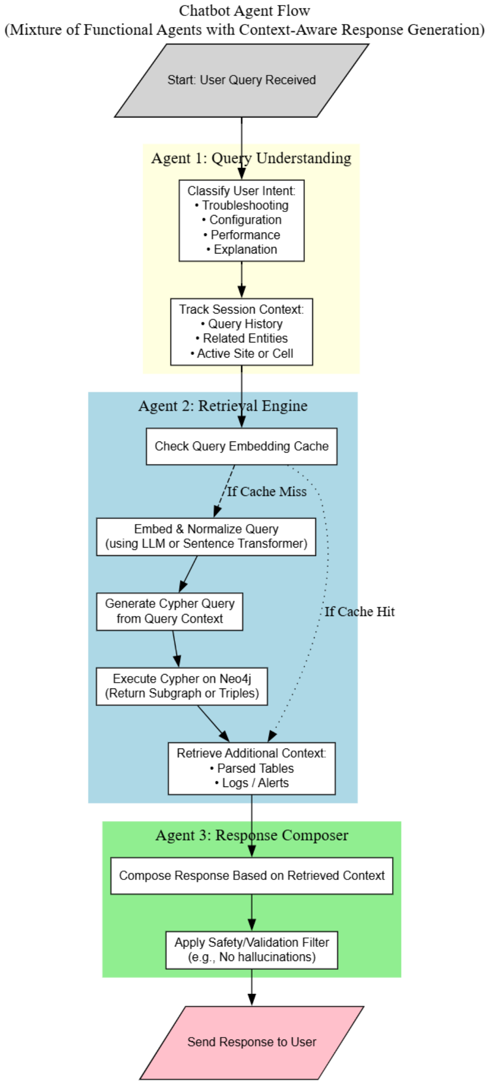
The technical implementation leverages Python and integrates several advanced libraries and frameworks. The Schema Describer Agent uses table fingerprints generated from DataFrame schemas to cache and reuse LLM-generated descriptions, as seen in the fingerprint_schema and describe_tables functions. This agent interacts with Fine-tuned Large Language Models, such as those accessed via the Google AI Studio API, to produce concise, context-aware descriptions of each table and its columns. Rate limiting and caching are handled robustly to ensure efficiency and reliability during repeated schema analysis.

Entity and Relation Extractor Agent further processes these descriptions, employing LLMs to extract core entities and their relationships from the tabular data. The extract_entities_and_relations function demonstrates how batching, caching, and robust JSON extraction are used to manage large outputs and avoid redundant LLM calls. This agent maintains an Entity-Relation Cache, enabling rapid retrieval of previously extracted patterns and supporting consistent modeling across diverse RAN data sources.

The Cypher Generator Agent translates the extracted entity-relation models into executable Cypher queries, as implemented in the build_cypher_commands function. These queries are then executed in a Neo4j graph database using the execute_cypher function, efficiently instantiating and updating the knowledge graph. The Cypher Query Cache allows for the reuse of previously generated queries, reducing computational overhead and improving response latency.

This functional arrangement ensures that the Knowledge Graph Constructor Agent can efficiently and incrementally update the knowledge graph within the Neo4j environment. The Mixture of Agent Architecture, combined with Fine-tuned Large Language Models and targeted caching strategies, enables the system to scale with growing data complexity, maintain semantic accuracy, and provide a robust foundation for advanced question answering and technical decision support in intelligent network management.

3.4.3. Chatbot Agent


Chatbot Agent Flow
(Mixture of Functional Agents with Context-Aware Response Generation)
Start: User Query Received
Agent 1: Query Understanding
Classify User Intent:
• Troubleshooting
• Configuration
• Performance
• Explanation
Track Session Context:
• Query History
• Related Entities
• Active Site or Cell
Agent 2: Retrieval Engine
Check Query Embedding Cache
If Cache Miss
Embed & Normalize Query
(using LLM or Sentence Transformer)
Generate Cypher Query
from Query Context
Execute Cypher on Neo4j
(Return Subgraph or Triples)
If Cache Hit
Retrieve Additional Context:
• Parsed Tables
• Logs / Alerts
Agent 3: Response Composer
Compose Response Based on Retrieved Context
Apply Safety/Validation Filter
(e.g., No hallucinations)
Send Response to User

3.5 Chatbot Agent Architecture Flow

The Chatbot Agent, as detailed in Figure 3.5, is a key enabler of Intelligent Network Management in Radio Access Networks. Its modular Mixture of Agent Architecture facilitates efficient, context-aware interactions with complex technical knowledge environments. The figure illustrates how specialized agents, each with a focused functional role, collectively deliver robust and adaptive query handling. By integrating both structured and unstructured data sources, the system ensures that responses are accurate, relevant, and grounded in the operational realities of RAN management.

At the core of the Chatbot Agent is the Query Understanding module, which combines semantic classification and context tracking to interpret user intent. This module leverages fine-tuned large language models, specifically adapted using RAN-specific datasets and knowledge graph-derived examples, to enhance its comprehension of technical queries and operational requirements. The fine-tuning process employs parameter-efficient techniques such as Low-Rank Adaptation (LoRA), which introduces a small set of trainable parameters into each transformer layer. This approach enables efficient domain adaptation with reduced computational cost and memory usage, making it practical to specialize large models for RAN management tasks. By applying LoRA-based fine-tuning on top of pre-trained models, the system becomes more sensitive to the terminology, data structures, and typical information needs encountered in RAN management, enabling more precise intent recognition and contextual interpretation.

The Intent Classification component discerns the high-level objective of each query such as troubleshooting, configuration inquiry, or performance monitoring by utilizing the fine-tuned model's improved understanding of domain-specific language and patterns. This allows downstream agents to tailor their retrieval and reasoning strategies to the user's actual needs. The Context Tracker maintains session continuity by monitoring historical queries, relevant network entities, and active RAN components, ensuring that each interaction is informed by the broader operational context. By integrating knowledge graph data into both the fine-tuning process and real-time context tracking, the Query Understanding module delivers robust, context-aware interpretations that drive accurate and relevant responses throughout the chatbot pipeline.

The Retrieval Engine agent orchestrates the gathering of contextual information necessary for precise response generation. Leveraging a Query Embedding Cache, the system efficiently identifies previously processed queries, while new queries are embedded and normalized using advanced language models. The architecture supports dual retrieval pipelines: Cypher Query Generation for translating intent and context into Neo4j-compatible queries, and Neo4j Query Execution for extracting relevant subgraphs or entity-relation triples from the knowledge graph. In parallel, the Contextual Retrieval component supplements responses with information from parsed datasets, system logs, and alert records, enabling hybrid reasoning across diverse data modalities.

Response synthesis is managed by the Response Composer agent, which assembles retrieved knowledge into coherent, contextually grounded replies. This process is guided by prompt templates or planning mechanisms and includes a Safety and Validation Check to ensure factual accuracy, consistency, and compliance with system policies. The modular, cache-optimized design of the Chatbot Agent supports real-time, adaptive dialogue and scalable deployment, while the integration of graph-based querying enables deep reasoning and traceable justifications, making it particularly well-suited for advanced technical support and decision-making in telecommunications network management.

3.5. Evaluation Method

The evaluation of the intelligent network management system is structured to assess both the functional performance of individual components and the overall effectiveness of the integrated architecture. The evaluation process is designed to ensure that the system meets the requirements for accuracy, efficiency, and usability in real-world RAN management scenarios. The following subsections outline the key evaluation methods and metrics used in this study.

3.5.1. Component Level Evaluation

Each module within the system is evaluated independently to verify its functionality and performance. This includes testing the Parser Module for its ability to accurately ingest and normalize diverse RAN data formats, assessing the Knowledge Graph Constructor Agent for its effectiveness in extracting entities and relationships, and evaluating the Chatbot Agent for its query understanding and response generation capabilities. Metrics such as parsing accuracy, entity extraction precision, and response relevance are employed to quantify performance at this level.

3.5.2. System Level Evaluation

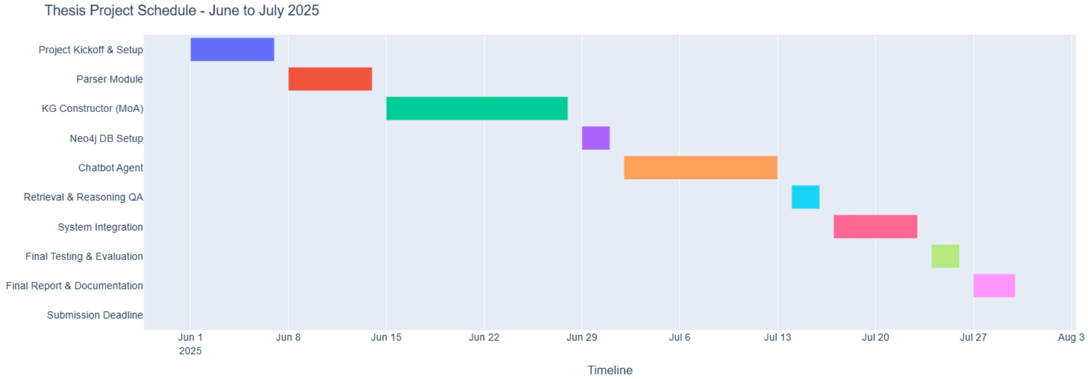
The integrated system is evaluated through a series of end-to-end tests that simulate real-world RAN management scenarios. These tests involve processing complex technical queries that require multi-step reasoning across different data sources. The evaluation focuses on metrics such as response time, accuracy of answers, and user satisfaction.

Additionally, the system's ability to handle concurrent queries and maintain context across interactions is assessed.

3.5.3. Comparative Evaluation

The performance of the proposed system is compared against baseline models and existing solutions in RAN management. This includes evaluating how well the fine-tuned LLMs perform relative to generic models, as well as comparing the MoA architecture's effectiveness against single-agent systems.

3.6. Schedule



3.6 Thesis Project Schedule

The project schedule is structured to ensure timely completion of each phase, with specific milestones for module development, integration, and evaluation. The timeline is divided into distinct phases, each with defined deliverables and deadlines. The schedule includes regular review points to assess progress and make necessary adjustments. The Gantt chart in Figure 3.6 illustrates the overall timeline, highlighting key activities such as data collection, module implementation, system integration, and evaluation.

REFERENCES

[1] Wang, J., Wang, J., Athiwaratkun, B., Zhang, C., & Zou, J. (2024, June 7). Mixture-of-Agents enhances large language model capabilities. arXiv.Org. https://arxiv.org/abs/2406.04692

[2] Lin, X. (2025). 3GPP Evolution from 5G to 6G: A 10-Year Retrospective. Telecom, 6(2), 32. https://doi.org/10.3390/telecom6020032

[3] Lee, Y. L., Qin, D., Wang, L.-C., & Sim, G. H. (2021). 6G massive radio access networks: Key applications, requirements and challenges. IEEE Open Journal of Vehicular Technology, 2, 54–66. https://doi.org/10.1109/ojvt.2020.3044569

[4] Zhou, H., Hu, C., Yuan, Y., Cui, Y., Jin, Y., Chen, C., Wu, H., Yuan, D., Jiang, L., Wu, D., Liu, X., Zhang, C., Wang, X., & Liu, J. (2024, May 17). Large language model (LLM) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities. arXiv.Org. https://arxiv.org/abs/2405.10825

[5] Erak, O., Alabbasi, N., Alhussein, O., Lotfi, I., Hussein, A., Muhaidat, S., & Debbah, M. (2024, August 21). Leveraging fine-tuned retrieval-augmented generation with long-context support: For 3GPP standards. arXiv.Org. https://arxiv.org/abs/2408.11775

[6] Parthasarathy, V. B., Zafar, A., Khan, A., & Shahid, A. (2024, August 23). The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. arXiv.Org. https://arxiv.org/abs/2408.13296

[7] Ethiraj, V., Vijay, D., Menon, S., & Berscilla, H. (2025, May 10). Efficient telecom specific LLM: TSLAM-Mini with qlora and digital twin data. arXiv.Org. https://arxiv.org/abs/2505.07877

[8] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., & Zhang, X. (2024, January 21). Large Language Model based Multi-Agents: A Survey of Progress and Challenges. arXiv.Org. https://arxiv.org/abs/2402.01680

[9] Wang, S., Fan, W., Feng, Y., Lin, S., Ma, X., Wang, S., & Yin, D. (2025, January 4). Knowledge graph retrieval-augmented generation for llm-based recommendation. arXiv.Org. https://arxiv.org/abs/2501.02226

[10] Jiang, P., Cao, L., Zhu, R., Jiang, M., Zhang, Y., Sun, J., & Han, J. (2025, February 16). RAS: Retrieval-And-Structuring for knowledge-intensive LLM generation. arXiv.Org. https://arxiv.org/abs/2502.10996

[11] Yuan, D., Zhou, H., Wu, D., Liu, X., Chen, H., Xin, Y., Jianzhong, & Zhang. (2025, March 31). Enhancing Large Language Models (LLMs) for Telecommunications using Knowledge Graphs and Retrieval-Augmented Generation. arXiv.Org. https://arxiv.org/abs/2503.24245

[12] Kumar, R., Ishan, K., Kumar, H., & Singla, A. (2025, March 11). LLM-Powered knowledge graphs for enterprise intelligence and analytics. arXiv.Org. https://arxiv.org/abs/2503.07993

[13] Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of Management Information Systems, 24(3), 45–77. https://doi.org/10.2753/mis0742-1222240302